

## Basi di dati

### Capitolo 6:

#### Progettazione di basi di dati: Metodologie e modelli per il progetto

1

## 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**

3/98

## Fasi (tecniche) del ciclo di vita

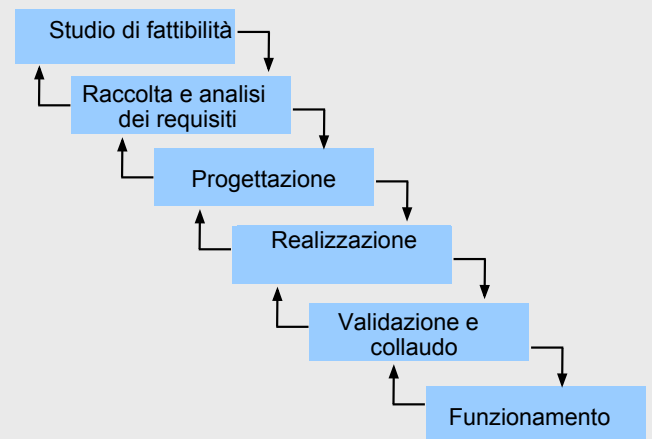
- Studio di fattibilità: definizione costi e priorità
- Raccolta e analisi dei requisiti: studio delle proprietà del sistema
- Progettazione: di dati e funzioni
- Realizzazione
- Validazione e collaudo: sperimentazione
- Funzionamento: il sistema diventa operativo

5/98

## Perché preoccuparci?

- 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.)
  - il modello relazionale è "rigido"

2/98



4/98

La progettazione di un sistema informativo riguarda due aspetti:

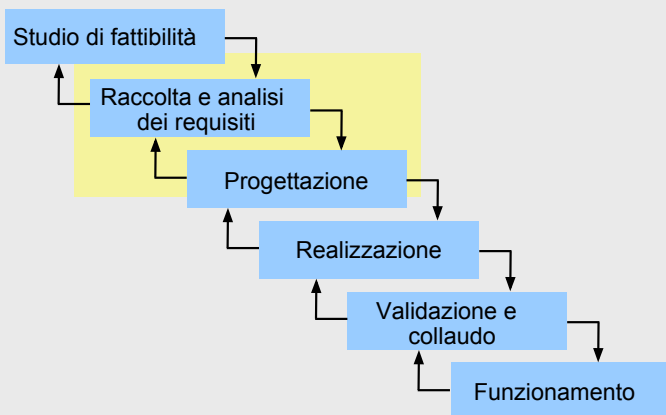
## progettazione dei dati progettazione delle applicazioni

### Ma:

i dati hanno un ruolo centrale

- i dati sono più stabili

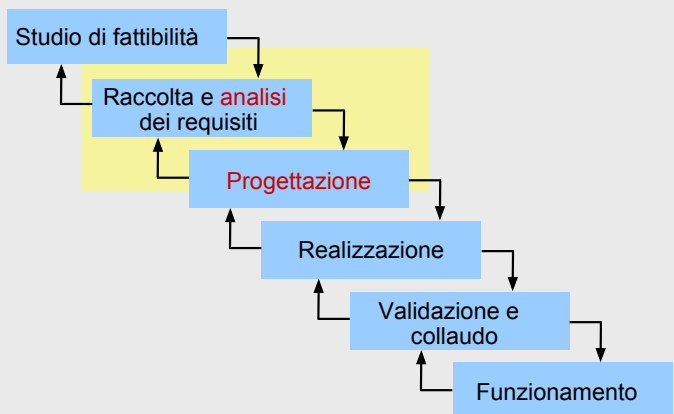
6/98



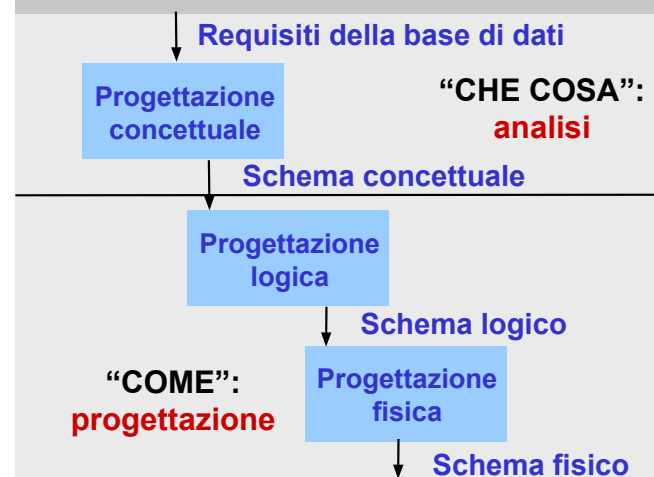
7/98

- Per garantire prodotti di buona qualità è opportuno seguire una
  - **metodologia di progetto**, con:
    - articolazione delle attività in fasi
    - criteri di scelta
    - modelli di rappresentazione
    - generalità e facilità d'uso

8/98



9/98



10/98

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

- Schema concettuale
- Schema logico
- Schema fisico

11/98

## Modello dei dati

- insieme di costrutti utilizzati per organizzare i dati di interesse e descriverne la dinamica
- componente fondamentale: **meccanismi di strutturazione** (o **costruttori di tipo**)
- come nei linguaggi di programmazione esistono meccanismi che permettono di definire nuovi tipi, così ogni modello dei dati prevede alcuni costruttori
- ad esempio, il **modello relazionale** prevede il costruttore **relazione**, che permette di definire insiemi di record omogenei

12/98

## Schemi e istanze

- In ogni base di dati esistono:
  - lo **schema**, sostanzialmente invariante nel tempo, che ne descrive la struttura (aspetto intensionale)
    - nel modello relazionale, le intestazioni delle tabelle
  - l'**istanza**, i valori attuali, che possono cambiare anche molto rapidamente (aspetto estensionale)
    - nel modello relazionale, il "corpo" di ciascuna tabella

13/98

## Due tipi (principali) di modelli

- modelli logici**: utilizzati nei DBMS esistenti per l'organizzazione dei dati
  - utilizzati dai programmi
  - indipendenti dalle strutture fisicheesempi: **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 progettazioneil più noto è il modello **Entità-Relazione** (nel seguito verrà chiamato modello **Entity-Relationship** per non confondersi con la relazione del modello relazionale)

14/98

## 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

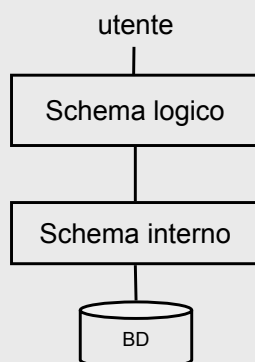
15/98

## Modelli concettuali, perché?

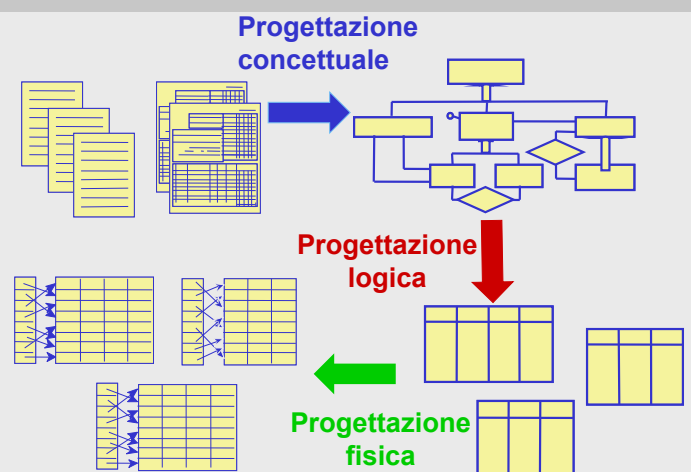
- servono per ragionare sulla realtà di interesse, indipendentemente dagli aspetti realizzativi
- permettono di rappresentare le classi di oggetti di interesse e le loro correlazioni
- prevedono efficaci rappresentazioni grafiche (utili anche per documentazione e comunicazione)

16/98

## Architettura (semplificata) di un DBMS



17/98



18/98

## Modello Entity-Relationship (Entità-Relazione)

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

19/98

## I costrutti del modello E-R

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

20/98

## Entità

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

21/98

## Relationship

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

22/98

## Uno schema E-R, graficamente



23/98

## Entità

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

24/98

## 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")

25/98

## Rappresentazione grafica di entità



26/98

## Entità, commenti

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

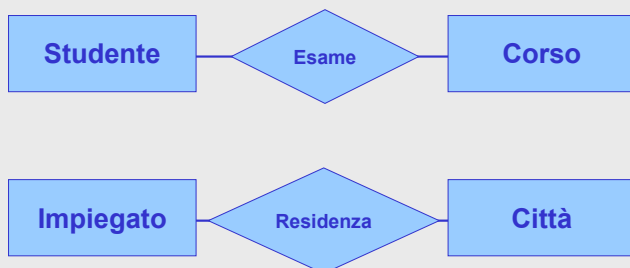
27/98

## 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**

28/98

## Rappresentazione grafica di relationship



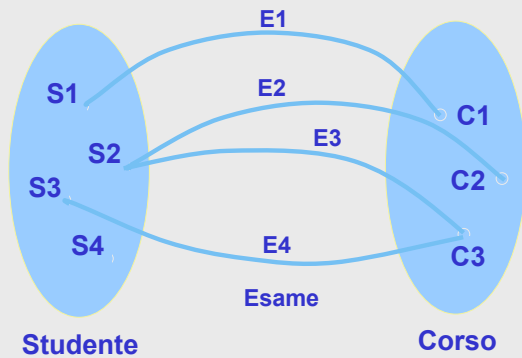
29/98

## Relationship, commenti

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

30/98

## Esempi di occorrenze



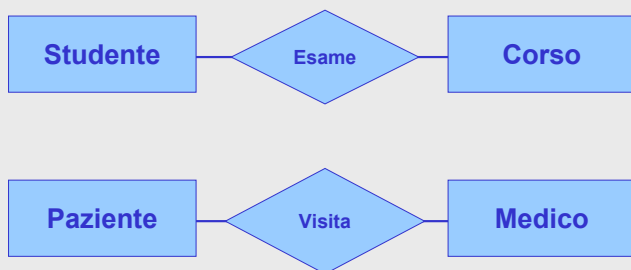
31/98

## 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

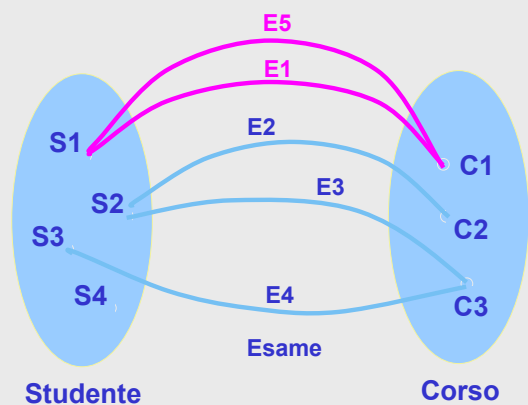
32/98

## Relationship corrette?



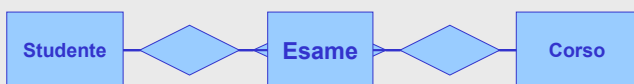
33/98

## Attenzione



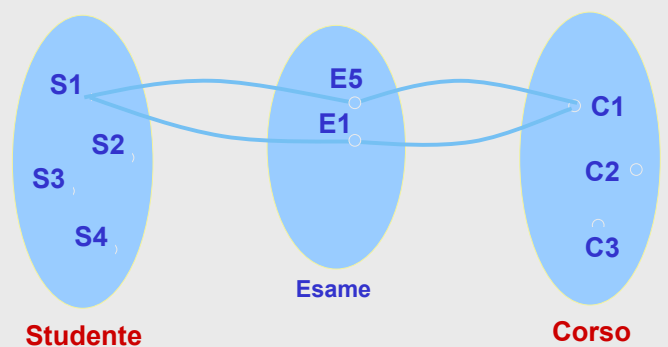
34/98

## "Promuoviamo" la relationship



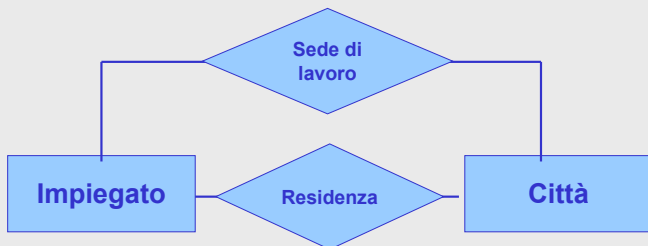
35/98

## Con l'entità Esame



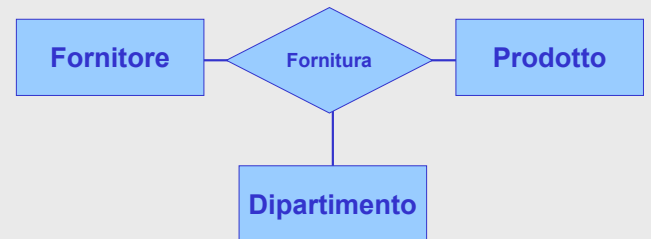
36/98

### Due relationship sulle stesse entità



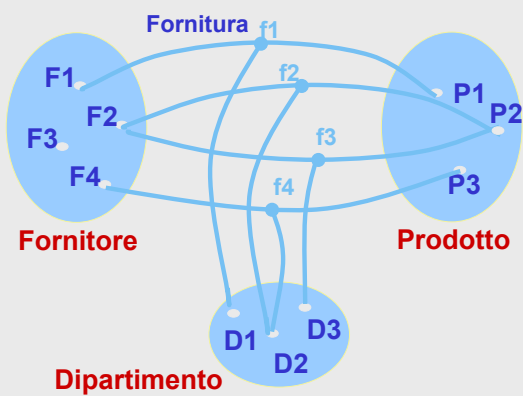
37/98

### Relationship n-aria



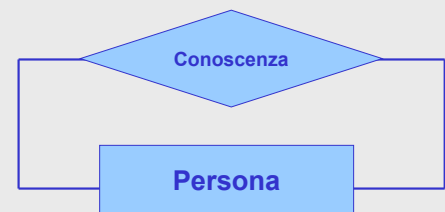
38/98

### Esempi di occorrenze



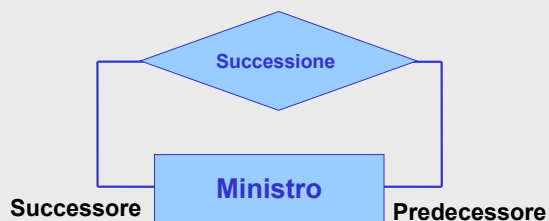
39/98

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



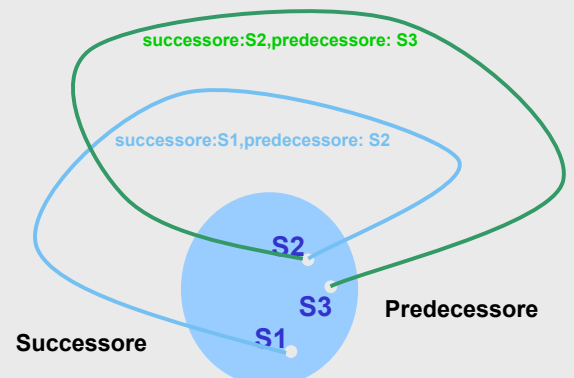
40/98

### Relationship ricorsiva con "ruoli"



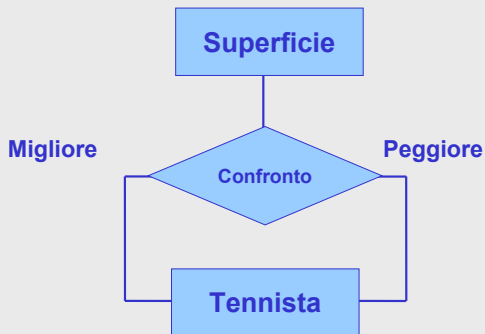
41/98

### Esempi di occorrenze



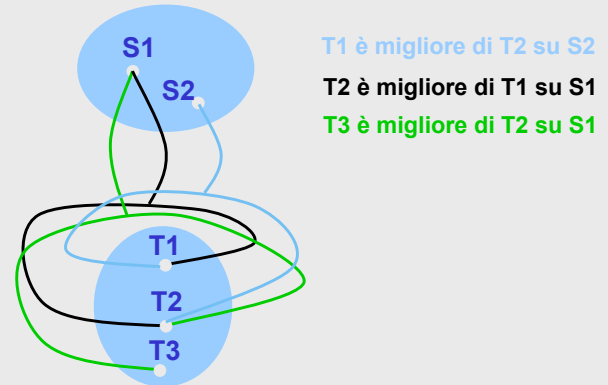
42/98

## Relationship ternaria ricorsiva



43/98

## Esempi di occorrenze



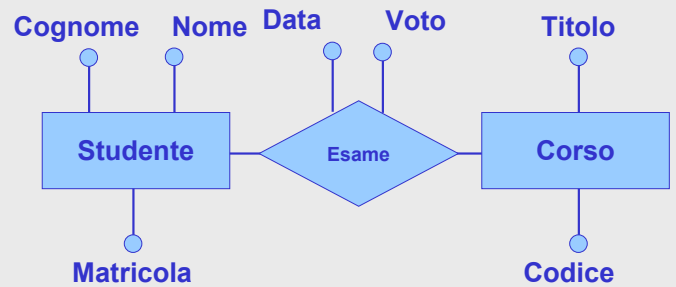
44/98

## 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

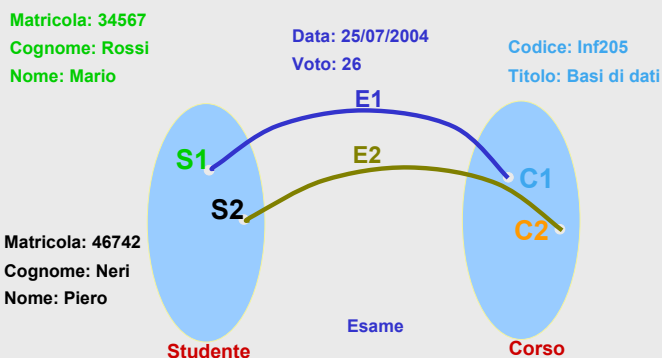
45/98

## Attributi, rappresentazione grafica



46/98

## Esempi di occorrenze



47/98

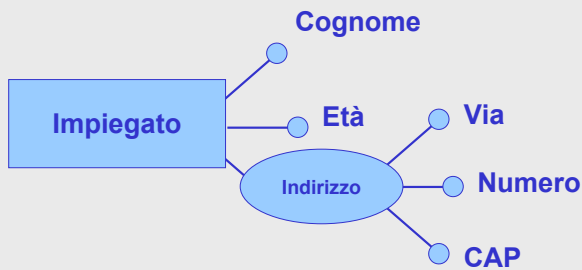
## Attributi composti

- 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

48/98



## Rappresentazione grafica



49/98

## Altri costrutti del modello E-R

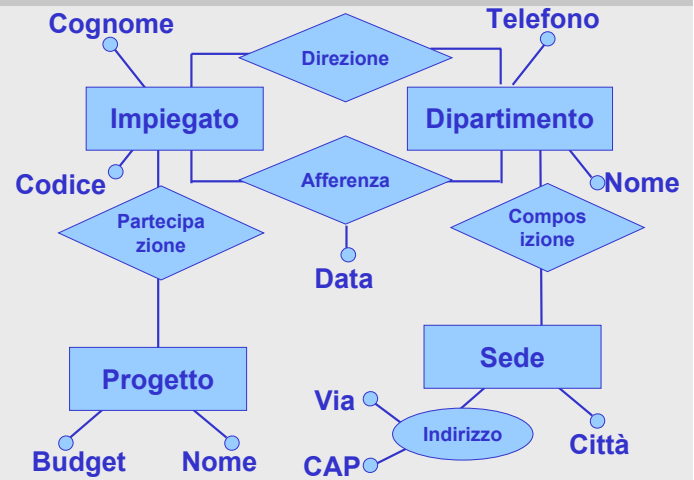
- Cardinalità
  - di relationship
  - di attributo
- Identificatore
  - interno
  - esterno
- Generalizzazione

51/98

## Esempio di cardinalità



53/98



50/98

## Cardinalità di relationship

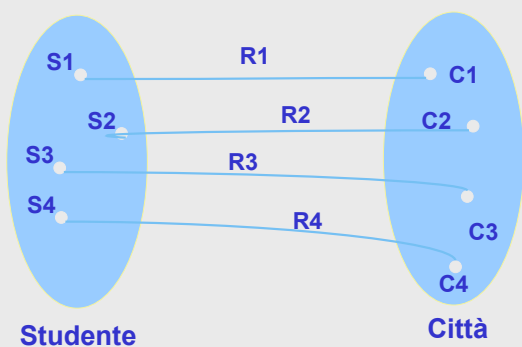
- Coppia di valori associati a ogni entità che partecipa a una relationship
- specificano il numero minimo e massimo di occorrenze delle relationship cui ciascuna occorrenza di una entità può partecipare

52/98

- 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

54/98

## Occorrenze di Residenza



55/98

## Cardinalità di Residenza



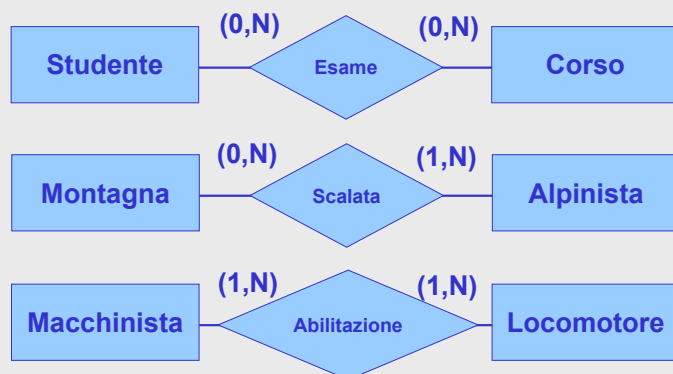
56/98

## Tipi di relationship

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

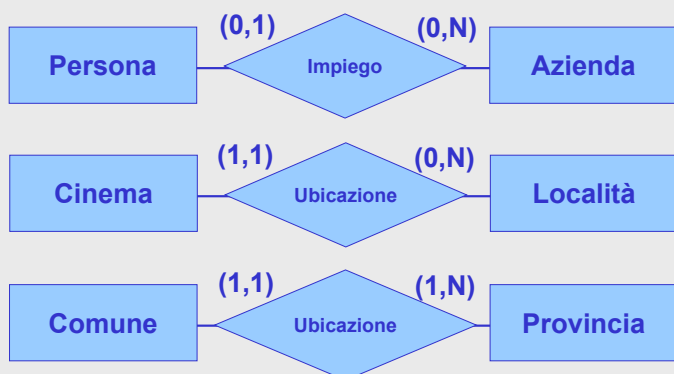
57/98

## Relationship “molti a molti”



58/98

## Relationship “uno a molti”



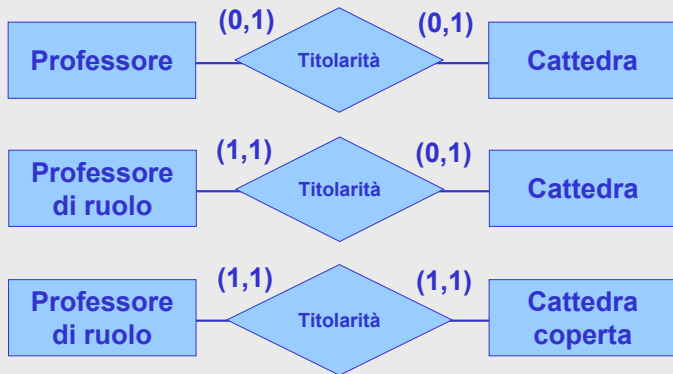
59/98

## Due avvertenze

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

60/98

### Relationship “uno a uno”



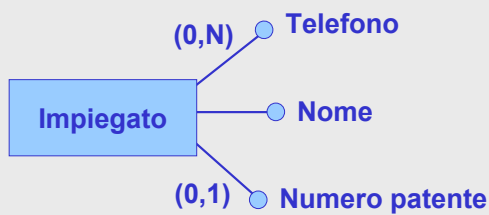
61/98

### Cardinalità di attributi

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

62/98

### Rappresentazione grafica



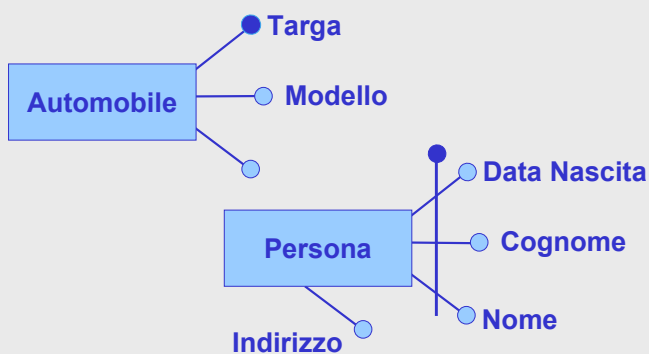
63/98

### 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

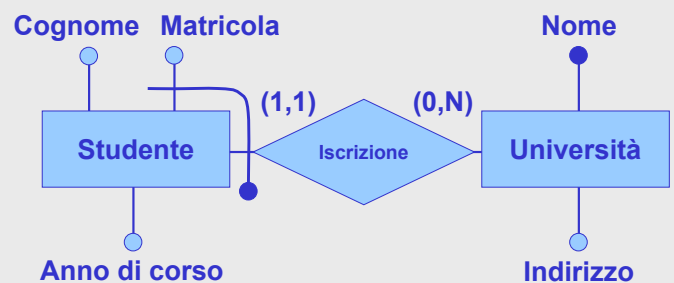
64/98

### Identificatori interni



65/98

### Identificatore esterno



66/98

## 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?

67/98

## Attenzione

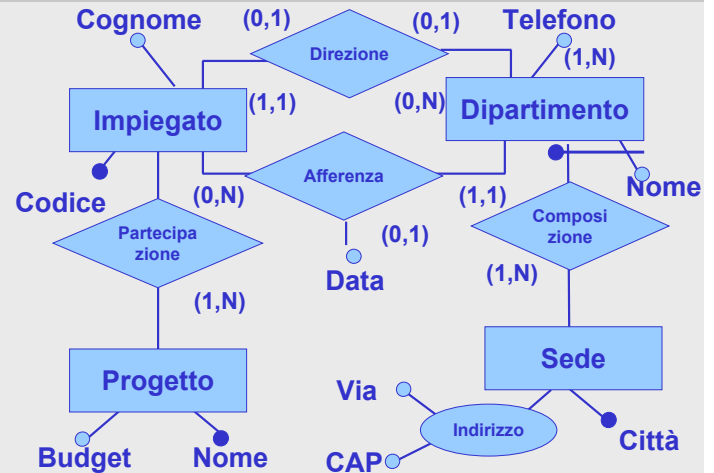
- Differenze apparentemente piccole in cardinalità e identificatori possono cambiare di molto il significato ...

69/98

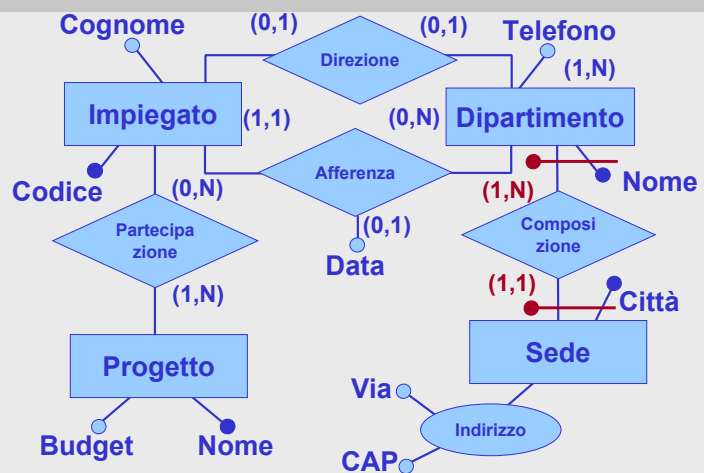
## Generalizzazione

- 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  $E$

71/98

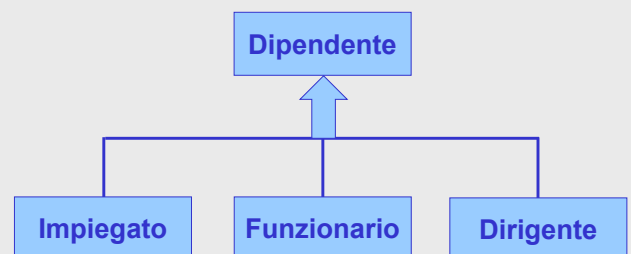


68/98



70/98

## Rappresentazione grafica



72/98

## Proprietà delle generalizzazioni

Se E (genitore) è generalizzazione di E1, E2, ..., En (figlie):

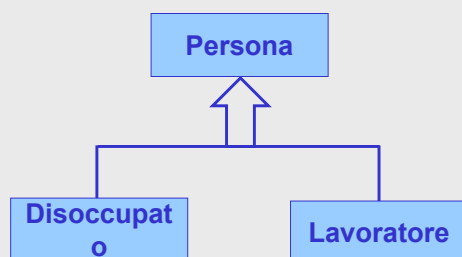
- ogni proprietà di E è significativa per E1, E2, ..., En
- ogni occorrenza di E1, E2, ..., En è occorrenza anche di E

73/98

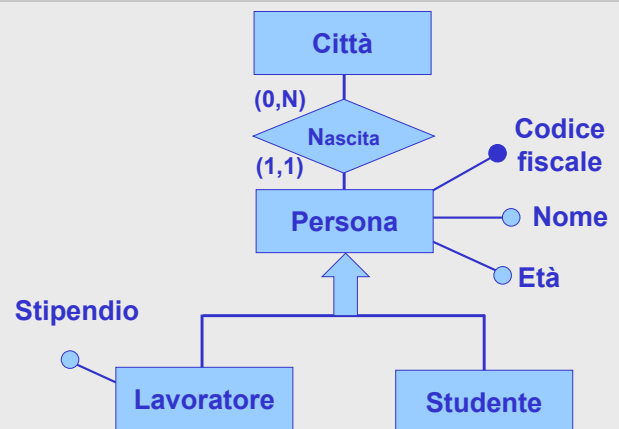
## Ereditarietà

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

75/98



77/98

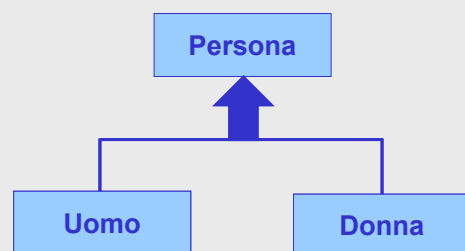


74/98

## 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

76/98

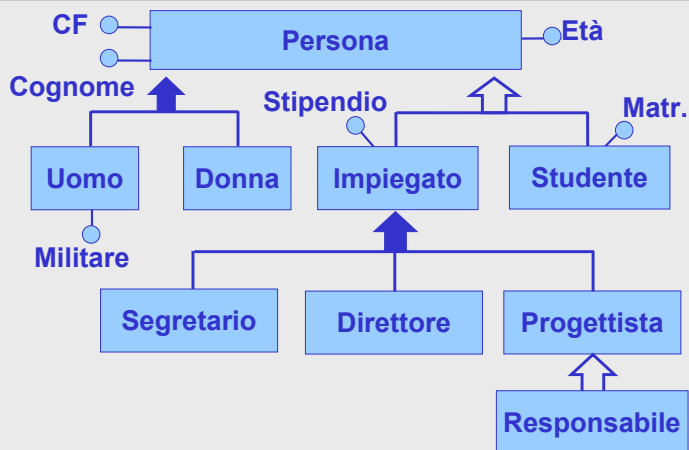


78/98

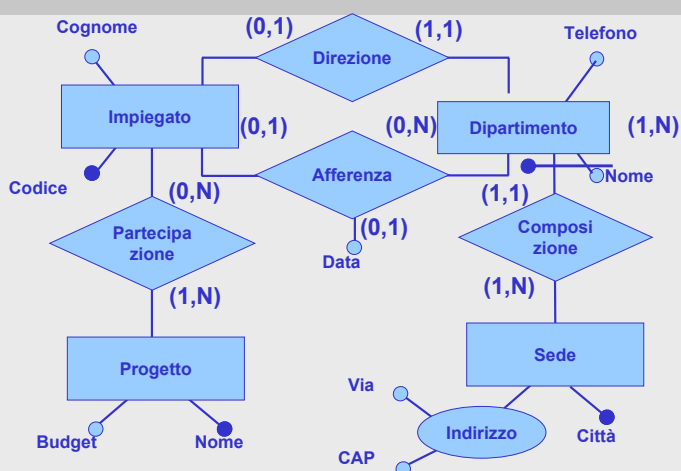
## Altre proprietà

- 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 ...

79/98



81/98



83/98

## 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)

80/98

## Documentazione associata agli schemi concettuali

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

82/98

## 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

84/98

## 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	

85/98

## 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

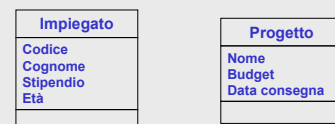
86/98

## Modellazione dei dati in UML

- UML viene talvolta utilizzato in alternativa al modello ER per la rappresentazione concettuale dei dati
- Si fa uso dei diagrammi delle classi
- Cambia la rappresentazione diagrammatica ma non l'approccio alla progettazione
- Vediamo come sia possibile rappresentare schemi concettuali con UML

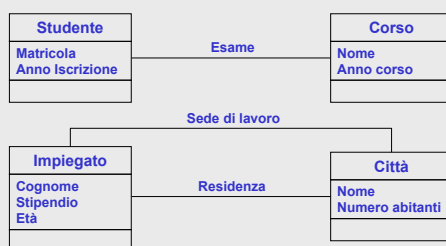
87/98

## Classi



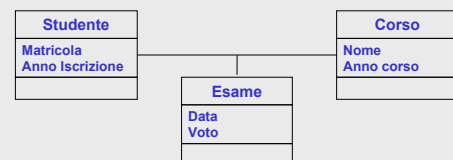
88/98

## Associazioni



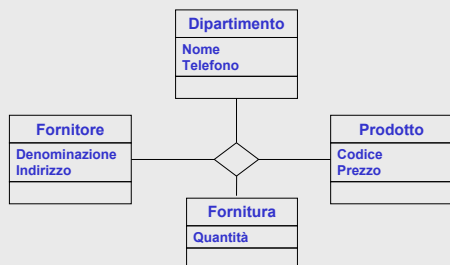
89/98

## Classe di associazione



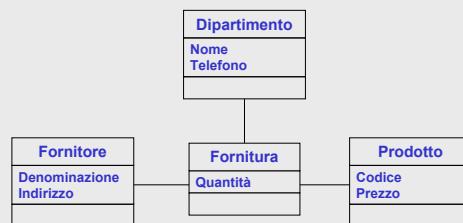
90/98

## Associazione ternaria



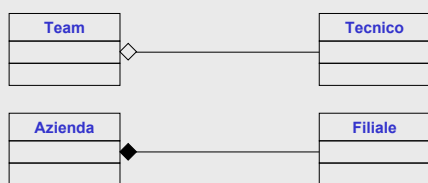
91/98

## Reificazione di associazione



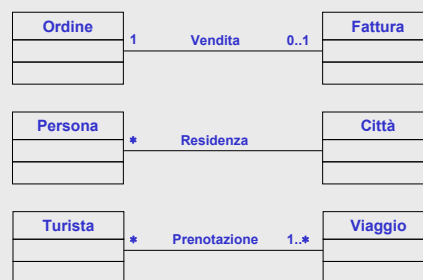
92/98

## Aggregazione e composizione



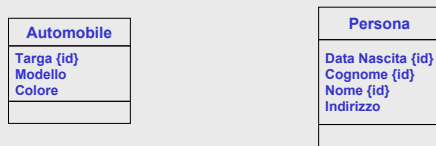
93/98

## Associazioni con molteplicità



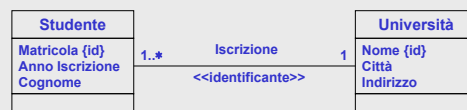
94/98

## Identificatori



95/98

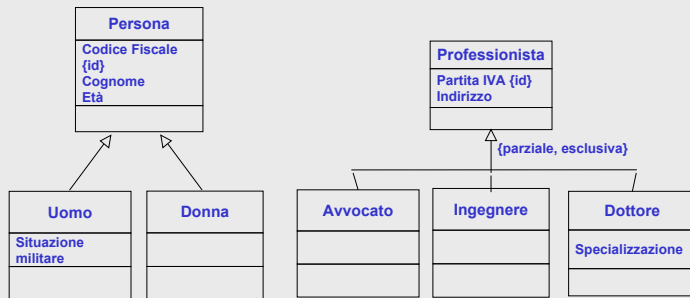
## Identificatore esterno



96/98

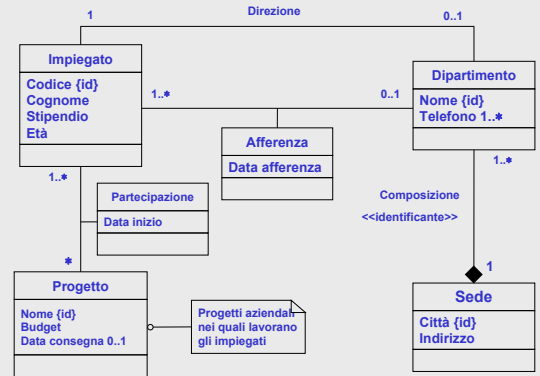


## Generalizzazioni



97/98

## Uno schema concettuale in UML



98/98