

Progettazione di basi di dati

Sistemi Informativi L

Corso di Laurea in Ingegneria dei Processi Gestionali
A.A. 2003/2004

Docente: Prof. Wilma Penzo



Progettazione di basi di dati

- È una delle attività del processo di sviluppo dei **sistemi informativi (SI)**
- 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**



Fasi del ciclo di vita di un SI

- Studio di fattibilità (determinazione dei costi e delle priorità)
- Raccolta e analisi dei requisiti
- Progettazione della base di dati e delle applicazioni
- Realizzazione
- Validazione e verifica del corretto funzionamento
- Operatività (normale funzionamento)
 - ...e altre (prototipazione, installazione, ecc.)



Ciclo di vita di un sistema informativo



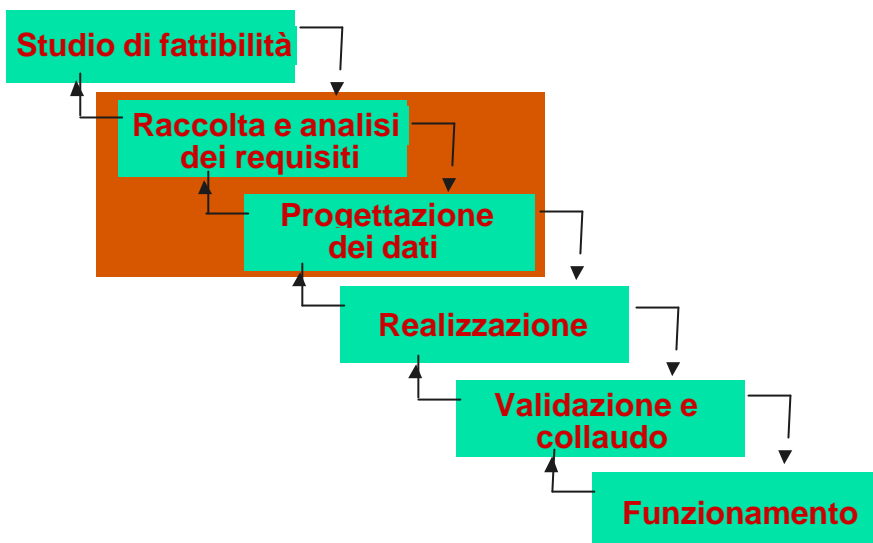


Progettazione guidata dai dati

- La progettazione di un sistema informativo riguarda 2 aspetti:
 - Progettazione dei dati
 - Progettazione delle applicazioni
- Il ruolo primario viene svolto dai dati, in quanto:
 - Sono (strutturalmente) più stabili
 - Sono condivisi da più applicazioni
- È quindi opportuno **progettare innanzitutto la base di dati** e successivamente le applicazioni



Ciclo di vita guidato dai dati





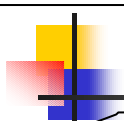
Metodologia di progettazione

- Per progettare una base di dati (ma non solo) di di buona qualità è opportuno seguire una

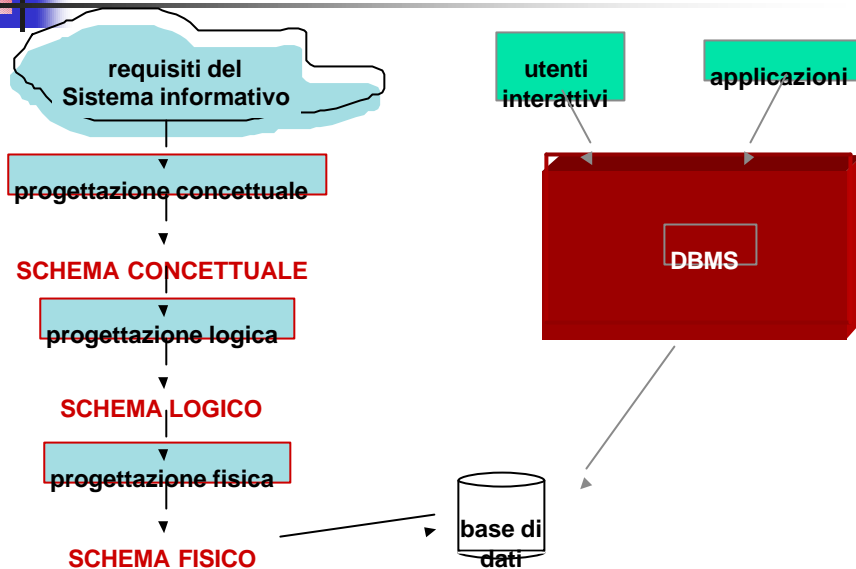
Metodologia di progettazione

che:

- definisca le fasi in cui l'attività di progettazione si articola
- fornisca dei criteri per scegliere tra diverse alternative
- sia supportata da opportuni modelli di rappresentazione
- sia di applicabilità generale e facile da utilizzare



Progettazione della base di dati





Fasi di progettazione

- La metodologia introdotta prevede 3 fasi:
 - **progettazione concettuale**
 - **progettazione logica**
 - **progettazione fisica**
- La fase di raccolta e analisi dei requisiti in pratica viene ad essere svolta congiuntamente a quella di progettazione concettuale
- Ognuna delle fasi si basa su un **modello**, che permette di generare una rappresentazione formale (**schema**) della base di dati ad un dato livello di astrazione (concettuale, logico e fisico):
 - **Schema concettuale**
 - **Schema logico**
 - **Schema fisico**



Fase di raccolta e analisi dei requisiti

- È la fase in cui si raccolgono e analizzano le specifiche informali ed eterogenee che i vari utenti danno delle procedure da automatizzare mediante un DBMS
 - **requisiti informativi**: caratteristiche dei dati
 - **requisiti sui processi**: operazioni sui dati
 - **requisiti sui vincoli di integrità**: proprietà dei dati e delle operazioni
- **Attività principali**:
 - Costruzione glossario dei termini
 - Eliminazione delle ambiguità (sinonimi, omonimi)
 - Raggruppamento dei requisiti “omogenei”
- Fase solo apparentemente semplice, nella realtà è spesso la più complessa perché è difficilmente standardizzabile il processo che porta a

capire cosa gli utenti vogliono!



Fase di progettazione concettuale

- A partire dai requisiti informativi viene creato uno **schema concettuale**, cioè una **descrizione formalizzata e integrata delle esigenze aziendali, espressa in modo indipendente dal DBMS adottato**
- A tale scopo si adotta un **modello concettuale**, che permette di fornire descrizioni ad alto livello indipendenti dall'implementazione
- Lo schema concettuale è indipendente anche dal **tipo** di DBMS che sarà utilizzato (relazionale, gerarchico, ecc.)



Fase di progettazione logica

- Consiste nella traduzione dello schema concettuale nel modello dei dati del DBMS
- Il risultato è uno **schema logico**, espresso nel DDL del DBMS
- In questa fase si considerano anche aspetti legati a:
 - integrità e consistenza (vincoli)
 - efficienza
- La progettazione logica si articola in due sotto-fasi:
 - ristrutturazione dello schema concettuale
 - traduzione verso il modello logico



Fase di progettazione fisica

- In questa ultima fase si operano scelte spesso strettamente dipendenti dallo specifico DBMS utilizzato
 - Ad esempio, lo stesso schema logico può essere fisicamente rappresentato in modo diverso in DB2 e in Oracle, al fine di meglio sfruttare le caratteristiche dei due DBMS
- Il risultato è lo **schema fisico**, che descrive le strutture di memorizzazione e accesso ai dati (tablespace, clustering, indici, ecc.)



Modelli dei dati: logici vs concettuali

Un modello dei dati è una collezione di concetti che vengono utilizzati per descrivere i dati, le loro associazioni, e i vincoli che questi devono rispettare

- Un ruolo di primaria importanza nella definizione di un modello dei dati è svolto dai **meccanismi che possono essere usati per strutturare i dati** (cfr. i costruttori di tipo in un linguaggio di programmazione)

Modelli logici: utilizzati nei DBMS esistenti per l'organizzazione dei dati

- utilizzati dai programmi, indipendenti dalle strutture fisiche

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

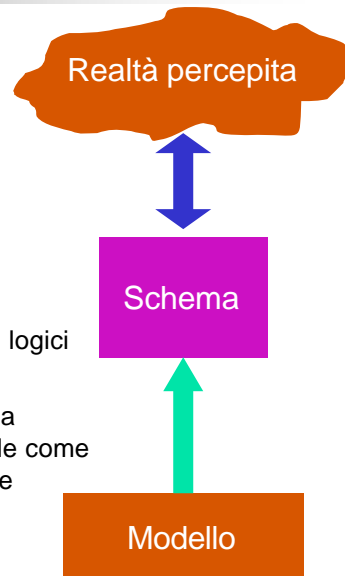
- Il più noto è il modello **Entity-Relationship**

Modelli concettuali... un po' di storia

- Nel tempo sono stati proposti numerosi modelli concettuali per la progettazione di basi di dati
 - modelli semantici, RM/T, ... [inizio anni '70]
 - **Entity-Relationship (E/R)** [entità-associazione] [Chen 1976]
 - IDEF1X [standard adottato dagli uffici governativi USA]
 - UML (Unified Modeling Language) [1999]

I modelli concettuali dei dati

- Lo scopo è pervenire a uno schema che rappresenti la realtà di interesse in modo indipendente dal DBMS
- Si cerca un livello di **astrazione** "intermedio" tra sistema e utenti, che sia al tempo stesso:
 - **Flessibile**
 - **Intuitivo**
 - **Espressivo**... tutte caratteristiche che mancano ai modelli logici
- I modelli concettuali prevedono tipicamente una rappresentazione grafica, che risulta anche utile come strumento di documentazione e comunicazione





Meccanismi di astrazione

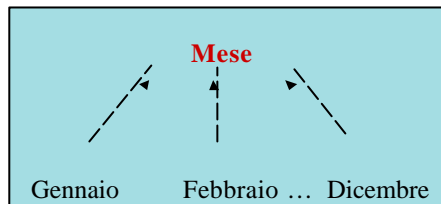
Concetto di astrazione: procedimento mentale che si adotta quando si concentra l'attenzione su alcune caratteristiche di un insieme di entità, trascurando le altre giudicate non rilevanti

- I meccanismi fondamentali di astrazione sono:
 - **classificazione**: identifica classi di oggetti del mondo reale aventi proprietà comuni
 - **aggregazione**: definisce un nuovo concetto a partire da concetti componenti
 - **generalizzazione**: definisce una classe astruendo dalle differenze esistenti tra due o più classi



Astrazione di classificazione

- Definizione di una classe a partire da un **insieme di oggetti aventi proprietà comuni**

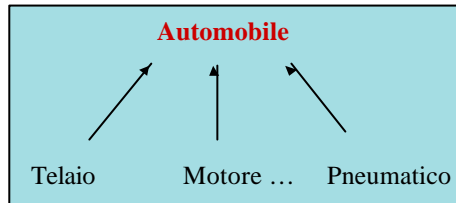


- Febbraio è un'**istanza** (instance, member) della classe Mese
- In generale, dato un insieme di oggetti, le classi definibili non sono univocamente determinate, dipende da ciò che interessa modellare!



Astrazione di aggregazione

- Definizione di un concetto a partire da un **insieme di concetti componenti**

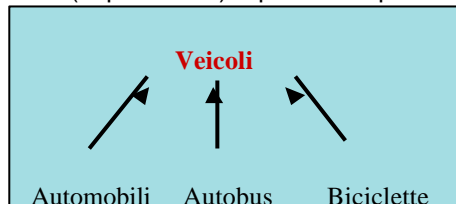


- Il Telaio è una **parte** (part of) dell'Automobile
- È la tipica astrazione che viene utilizzata quando si definiscono dei record (tuple)



Astrazione di generalizzazione

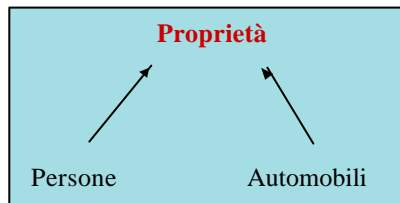
- Definisce una classe (superclasse) a partire da più classi (sottoclassi)



- Le istanze di Automobili sono un sottoinsieme delle istanze di Veicoli, ovvero, ogni automobile **è un** (is a) veicolo
- Ciò che caratterizza un veicolo caratterizza anche ogni suo sottoinsieme, ovvero ogni sottoclasse **eredita** dalla superclasse...
- ... ma può anche avere caratteristiche proprie
- La specializzazione è il processo inverso

Corrispondenze tra classi

- Oltre alla generalizzazione è possibile stabilire altre corrispondenze tra classi; tali corrispondenze sono di fatto delle aggregazioni, di cui le classi sono le componenti



- Dal punto di vista della modellazione, è importante caratterizzare queste corrispondenze in termini di **vincoli di cardinalità** (un'automobile quanti proprietari può avere?)

Vincoli di cardinalità

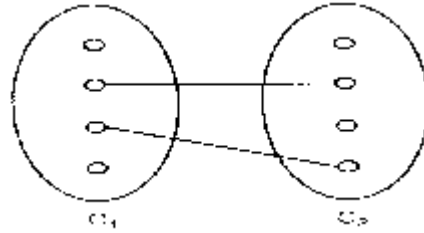
- Sia **A** aggregazione di **C1** e **C2**
 - **min-card(C1,A)**: cardinalità minima di C1 in A
 - è il minimo numero di corrispondenze nell'aggregazione A alle quali ogni istanza di C1 deve partecipare
 - **max-card(C1,A)**: cardinalità massima di C1 in A
 - è il massimo numero di corrispondenze nell'aggregazione A alle quali ogni istanza di C1 può partecipare
- **Vincoli di cardinalità minima**
 - **partecipazione opzionale**: **min-card(C1,A) = 0**
 - alcuni elementi di C1 possono non essere aggregati tramite A a elementi di C2
 - **partecipazione obbligatoria (totale)**: **min-card(C1,A) > 0**
 - ad ogni elemento di C1 deve essere aggregato, tramite A, almeno un elemento di C2



Vincoli di cardinalità massima (1)

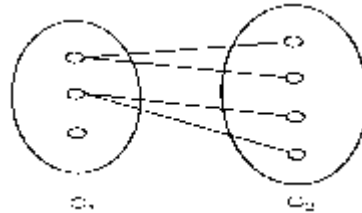
- **Uno a uno (One-to-one)**

$\max\text{-card}(C1,A)=1$
 $\max\text{-card}(C2,A)=1$



- **Uno a molti (One-to-many)**

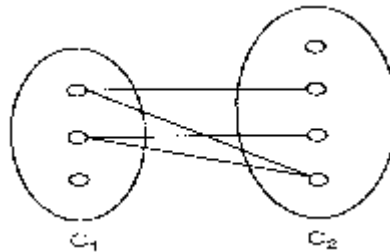
$\max\text{-card}(C1,A)=n$
 $\max\text{-card}(C2,A)=1$



Vincoli di cardinalità massima (2)

- **Molti a molti (Many-to-many)**

$\max\text{-card}(C1,A)=n$
 $\max\text{-card}(C2,A)=n$

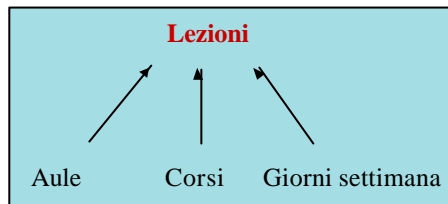


$\max\text{-card}(C1,A)=n$ significa che il massimo numero di partecipazioni di un elemento di C1 in A è limitato soltanto dai numeri di elementi in C2



Aggregazione n-aria

- Quanto visto si estende naturalmente al caso di n classi



Ogni corso ha non più di 3 lezioni settimanali

min-card(Corsi,Lezioni)=1; max-card(Corsi,Lezioni)=3

- Ragionare sui vincoli è in generale meno immediato che nel caso binario

Ogni corso si tiene in non più di 2 aule

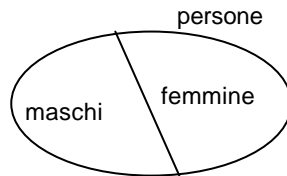
- Non riguarda i giorni, quindi non riguarda Lezioni!



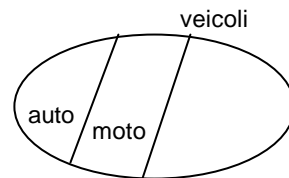
Copertura delle generalizzazioni

- Le generalizzazioni si caratterizzano per due dimensioni indipendenti
- **Confronto fra unione delle specializzazioni e classe generalizzata**
 - **totale** se la classe generalizzata è l'unione delle specializzazioni
 - **parziale** se la classe generalizzata contiene l'unione delle specializzazioni
- **Confronto fra le classi specializzate**
 - **esclusiva** se gli insiemi delle specializzazioni sono fra loro disgiunti
 - **sovrapposta (overlapped)** se può esistere una intersezione non vuota fra insiemi delle specializzazioni
- Sono ovviamente possibili le quattro combinazioni
(t,e) (p,e) (t,o) (p,o)

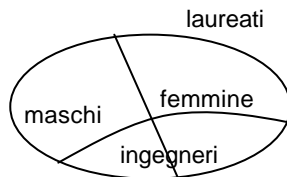
Proprietà di copertura - esempi



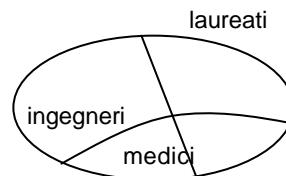
Totale esclusiva



Parziale esclusiva



Totale sovrapposta



Parziale sovrapposta

Conclusioni

- La **progettazione di un sistema informativo** è guidata dai dati, e si avvale di una **metodologia** che consta di diverse fasi
- Ogni fase produce uno **schema**, facendo uso di uno specifico **modello**
- Per la **progettazione concettuale** si fa uso di un **modello concettuale** che astrae da aspetti specifici dei DBMS
- Tutti i modelli concettuali si basano su alcuni **meccanismi di astrazione** fondamentali:
 - **Classificazione**
 - **Aggregazione**
 - **Generalizzazione**