FASI NEL PROGETTO DEI DATI

Il progetto della base di dati

si inserisce nel:

Ciclo di vita del sistema informativo

comprendente in generale le seguenti attività:

- **Raccolta ed analisi dei requisiti**
- **Progettazione (di schemi e applicazioni)**
- **Implementazione**
- **Validazione e collaudo**
- **Funzionamento**

Ci concentriamo sulla parte più tecnica, specifica di questo corso: la progettazione degli schemi

Prerequisiti della progettazione

- L'analisi dei requisiti è condotta dalla una specifica figura professionale (analista) tramite interviste dell'utente (che deve essere per quanto possibile parte attiva della definizione dei requisiti)
- Produce una descrizione che esprimiamo tramite un testo riassuntivo (tipicamente ambiguo).
- Nella realtà si aggiungono anche:
  - Descrizioni terminologiche, glossari, raccolte informali di definizioni
  - Descrizioni astratte dei programmi (dataflow diagrams) e del loro uso da parte degli utenti (use case diagrams).

Assunzione tecnologica

- Architettura client-server con un unico database server cui si collegano le varie applicazioni

(di fatto questa scelta non è vincolante e architetture pù complesse sono discusse nell'ambito del corso di Sistemi Informativi)

Fasi della progettazione

- **la progettazione concettuale**
- **la progettazione logica**
- **la progettazione fisica**

La progettazione concettuale

Ha per scopo tradurre il risultato dell'analisi dei requisiti in una DESCRIZIONE FORMALE che dovrà essere indipendente dal DBMS

La descrizione formale è espressa tramite uno SCHEMA CONCETTUALE, costruito utilizzando un MODELLO CONCETTUALE DEI DATI

La progettazione logica

Ha per scopo tradurre lo SCHEMA CONCETTUALE in uno SCHEMA LOGICO, scelto all'interno dei modelli logici dei dati:

- **Gerarchico**
- **Reticolare**
- **Relazionale**
- **Orientato ad oggetti**
- **XML**

Lo schema logico è dipendente dal DBMS ma non dallo specifico prodotto

La progettazione fisica

Ha per scopo produrre un PROGETTO FISICO della base dei dati, cioè un progetto che ottenga prestazioni ottimali tramite scelta e dimensionamento di strutture fisiche di accesso.

Il progetto fisico viene eseguito in modo differente su ciascun prodotto.

Dipendenze da MODELLO e DBMS

	Dipende dal MODELLO	Dipende dal DBMS
Progetto concettuale	NO	NO
Progetto logico	SI	NO
Progetto fisico	SI	SI

LE ASTRAZIONI NEI MODELLI CONCETTUALI PER BASI DI DATI

Ingredienti dei modelli concettuali

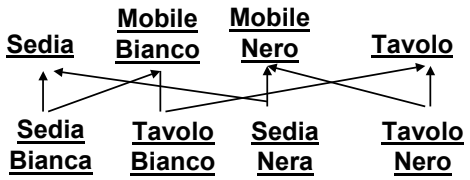
- ASTRAZIONI: capacità di evidenziare caratteristiche comuni ad insiemi di oggetti
- Tre ASTRAZIONI di base per la rappresentazione della conoscenza:
  - Classificazione
  - Aggregazione
  - Generalizzazione

Classificazione

Capacità di definire classi di oggetti o fatti del mondo reale

- LIBRO
- BICICLETTA
- PERSONA
- APPARTAMENTO

Per ogni classe esiste un implicito "test di appartenenza" che consente di dire se un oggetto o fatto del mondo reale è una istanza della classe.

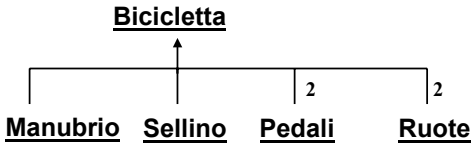


Aggregazione

Costruzione di una classe complessa aggregando classi più semplici (componenti)

- BICICLETTA
  - RUOTE
  - PEDALI
  - MANUBRIO

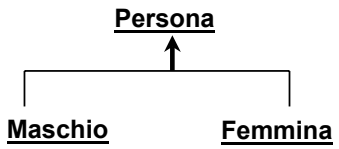
Per ogni componente si indica quante istanze sono presenti in una istanza della classe aggregata



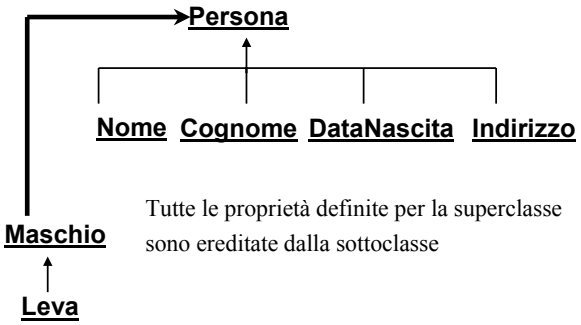
Generalizzazione

Stabilisce legami di sottoinsieme fra classi

- FEMMINA < PERSONA
- MASCHIO < PERSONA



Ereditarietà



Tutte le proprietà definite per la superclasse sono ereditate dalla sottoclasse

IL MODELLO ENTITA'-RELAZIONE (ENTITY-RELATIONSHIP)

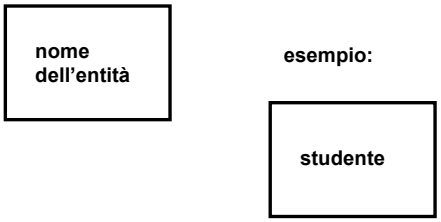
MODELLO ENTITA' RELAZIONE

- il modello Entity-Relationship (ER), (P.P.Chen 1976) si è affermato come standard industriale di buona parte delle metodologie e degli strumenti per il progetto concettuale di basi di dati.
- Attenzione: Relationship = Associazione (pero' poi si dice informalmente "relazione")

Entità

- Rappresenta una classe di oggetti (es., automobili, impiegati, studenti) o di fatti (es., conti correnti, corsi universitari)
- Devono essere oggetti rilevanti per la applicazione
- Ogni entità è caratterizzato da un nome

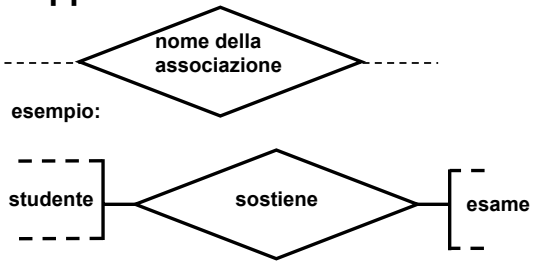
simbolo grafico per rappresentare entità



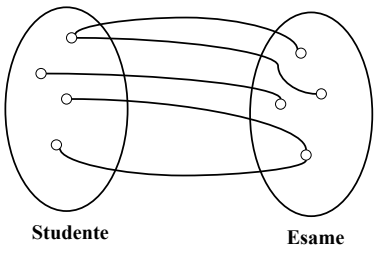
Relazione (o Associazione)

- Rappresenta una aggregazione di entità di interesse per l'applicazione
- Ogni istanza di una associazione è una ennupla tra istanze di entità (es., legame tra un automobile e il suo proprietario)
- Ogni associazione è caratterizzata da un nome

simbolo grafico per rappresentare associazioni



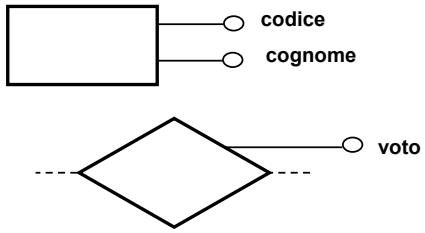
Rappresentazione grafica delle istanze



Attributi

- Rappresentano caratteristiche delle entità e delle associazioni di interesse per l'applicazione
- Ogni attributo è caratterizzata da un nome

simbolo grafico per rappresentare attributi



Linee guida per il progetto

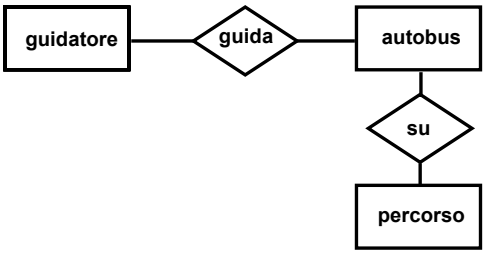
- Se il concetto è significativo per il contesto applicativo: entità
- Se il concetto è descrivibile tramite un dato elementare: attributo
- Se il concetto definisce un legame tra entità: associazione

Corrispondenza tra concetti e elementi del modello ER

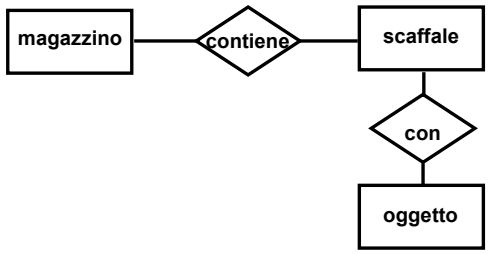
La corrispondenza tra oggetti e fatti del mondo reale e entità, associazioni e attributi non è assoluta ma dipende dal contesto:

Es.: l'auto BOF34675 ha colore rosso
il colore rosso ha lunghezza d'onda = ~700 nm

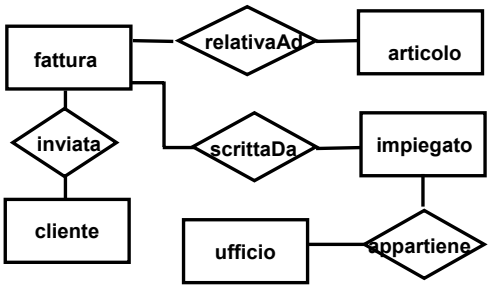
Esempio: gestione viaggi



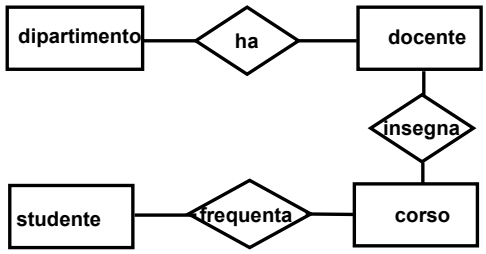
Esempio: gestione magazzino



Esempio: gestione fatture



Esempio: università



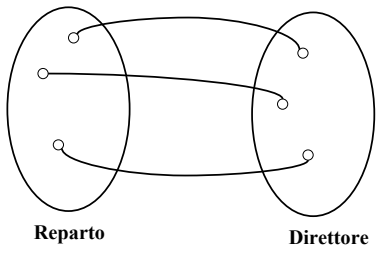
ASSOCIAZIONI NEL MODELLO ER

Cardinalità delle associazioni

- Per cardinalità si intende un vincolo sul numero di istanze di associazione cui ciascuna istanza di entità deve partecipare.
- È una coppia (MIN-CARD, MAX-CARD)
  - MIN-CARD = 0 (opzionale)
  - = 1 (obbligatoria)
  - MAX-CARD = 1 (uno)
  - = N (molti)
- In base alla sola cardinalità massima si hanno associazioni uno-uno, uno-molti, molti-molti

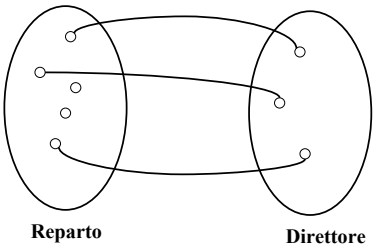
Associazione 1:1

- un reparto deve essere diretto da uno e un solo direttore (1,1)
  un direttore deve dirigere uno ed un solo reparto (1,1)



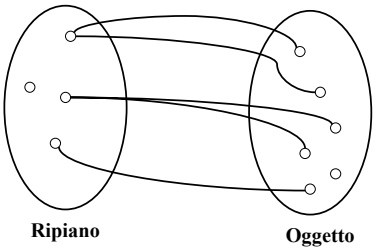
Associazione 1:1 con opzionalità

- un reparto puo' essere diretto da uno e uno solo direttore (0,1)
  un direttore deve dirigere uno ed un solo reparto (1,1)



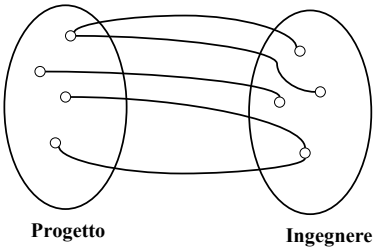
Associazione 1:N

- un reparto puo' contenere molti oggetti (0,n)
  un oggetto può essere contenuto al più su un ripiano (O,1)



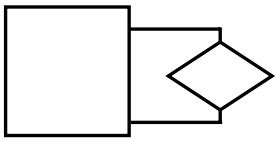
Associazione N:M

- un progetto puo' essere fatto da molti ingegneri (0,n),
  un ingegnere deve partecipare ad uno o più progetti (1,m)

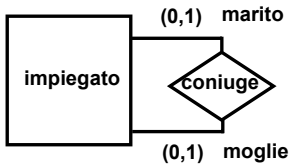


Auto-associazioni

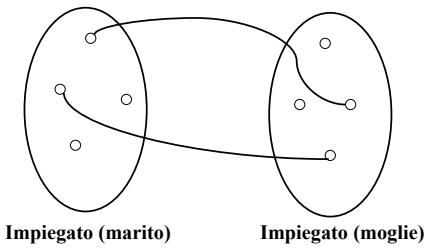
associazioni aventi come partecipanti istanze provenienti dalla stessa entità (chiamate anche "ad anello"):



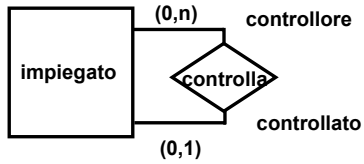
Auto-associazioni 1:1



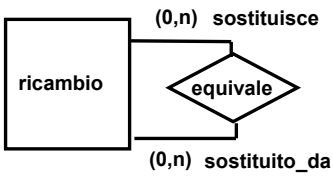
può essere riportato il "ruolo" sul ramo



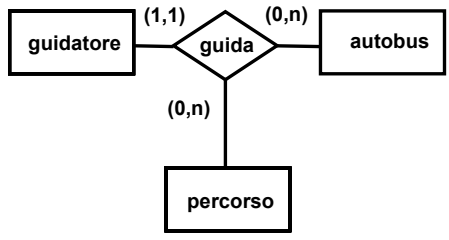
Auto-associazioni 1:N

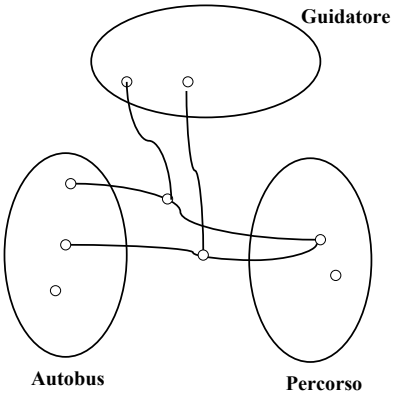


Auto-associazioni N:M

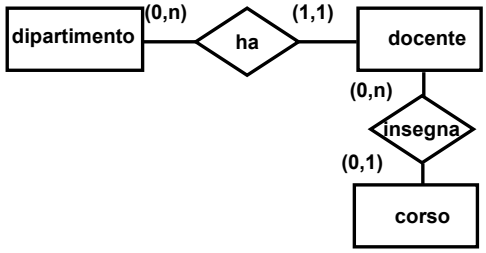


Associazioni ternarie

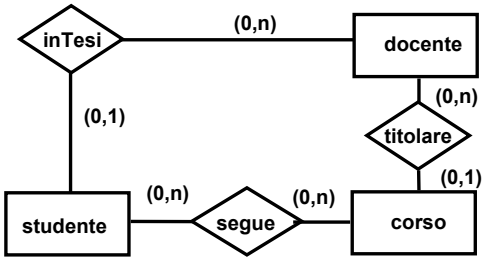




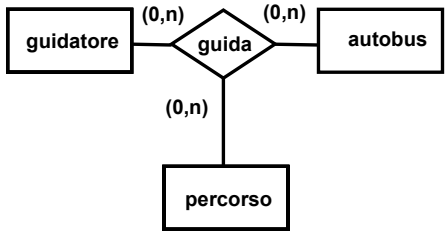
Esempio:università



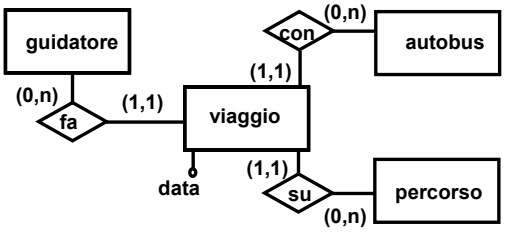
Esempio: università



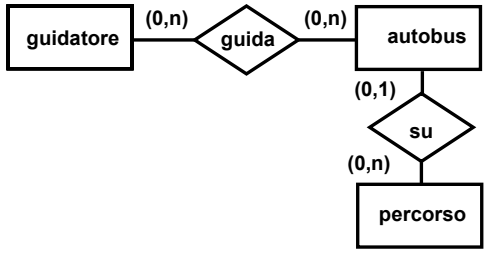
Esempio: gestione autobus



Esempio: gestione autobus



Esempio: gestione autobus



ATTRIBUTI E IDENTIFICATORI NEL MODELLO ER

Cardinalità degli attributi

- una prima classificazione:

attributo scalare (semplice, ad un sol valore)

es.: matricola, cognome, voto

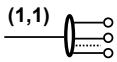
attributo multiplo (sono ammessi n valori)

es.: qualifica, titolo, specialità

il simbolo (n,m) esprime la cardinalità dell'attributo.

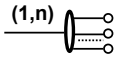
Attributi composti

attributo composto



es.: data (gg,mm,aaaa), indirizzo (via, numero civico, città, provincia, cap)

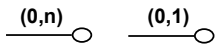
attributo multiplo composto



es.: telefono (stato, città, numero)

Opzionalità

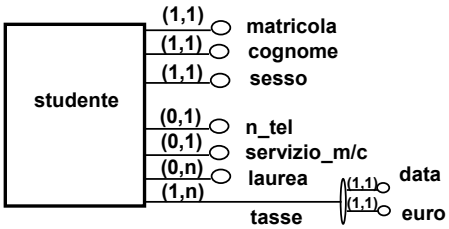
Attributo opzionale (è ammessa la "non esistenza del valore")



es.: tel., qualifica, targa

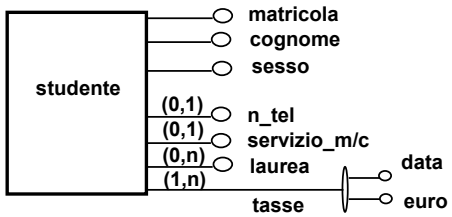
Esempio

Esempio:



Esempio con default

Esempio:

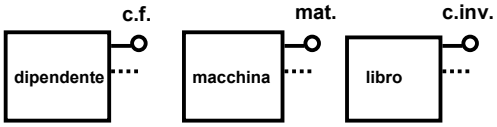


Identificatore

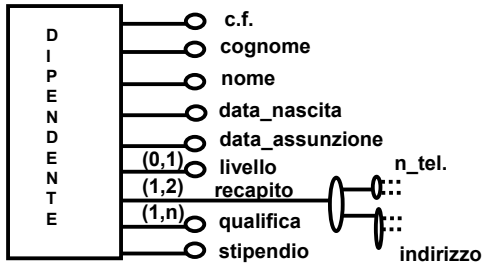
Un identificatore caratterizza in modo univoco ciascuna singola istanza di entità

simbolo

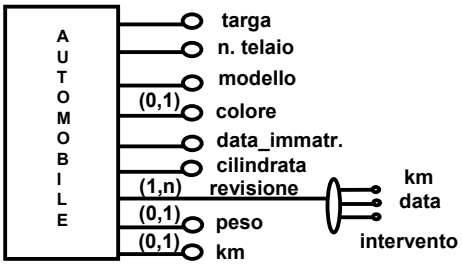
non è modificabile (in generale…)



esempio

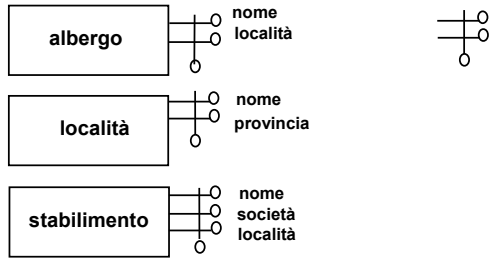


esempio

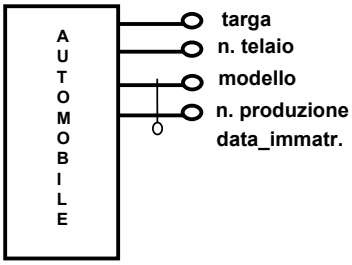


Identificatori composti

L'identificatore di un'entità può essere composto



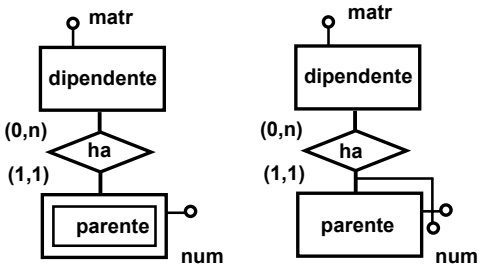
esempio



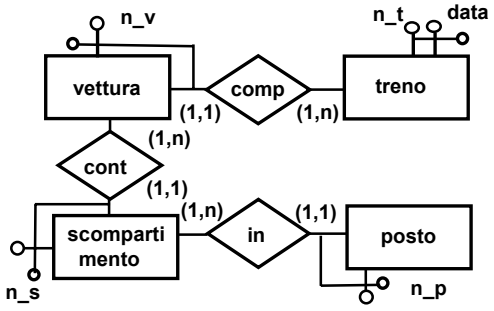
Le entità deboli

- **Le entità deboli possono esistere se e solo se sono presenti entità "forti" da cui queste dipendono**
  - **In caso di eliminazione dell'istanza "forte" di riferimento le istanze deboli collegate devono essere eliminate**

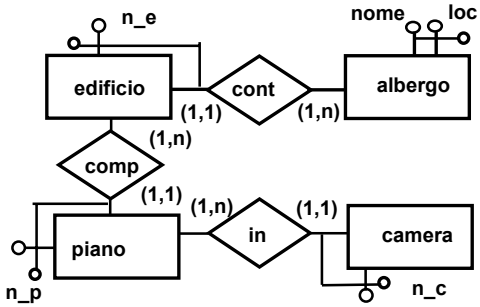
Simboli usati



esempio: posti treno



Esempio: gestione camere di un albergo



Esempi: commento

- **le entità con identificatore esterno sono deboli poiché a tutti i livelli la cancellazione di una entità provoca la cancellazione delle entità deboli collegate**

  (eliminazione di vettura, chiusura di scompartimento, inagibilità edificio o piano, ecc.)

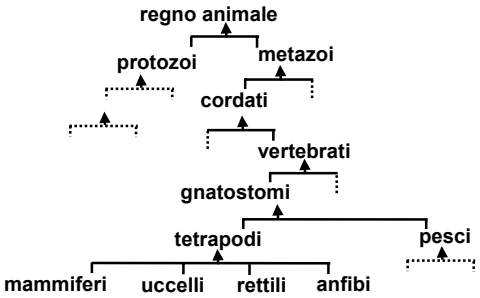
GERARCHIE DI GENERALIZZAZIONE

Gerarchie di generalizzazione

- **Una gerarchia di generalizzazione è un legame logico tra un'entità padre E ed alcune entità figlie E_1 E_2 .. E_n dove:**
  - **E è la generalizzazione di E_1 E_2 .. E_n**
  - **E_1 E_2 .. E_n sono specializzazioni di E**

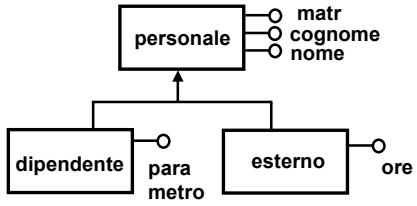
  tale per cui:
  - **ogni istanza di E_k è anche istanza di E**
  - **una istanza di E può essere una istanza di E_k**
- **Le entità figlio ereditano le proprietà (attributi, relazioni, identificatori) dell'entità padre.**

Una delle gerarchie più note



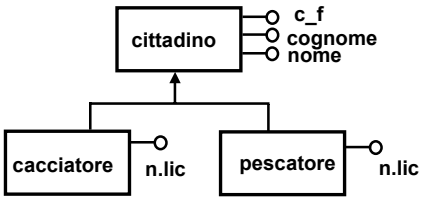
Esempio di gerarchia

un'azienda si avvale dell'opera di professionisti esterni, quindi il suo personale si suddivide in esterni e dipendenti:



Esempio di gerarchia

un comune gestisce l'anagrafe ed i servizi per i suoi cittadini alcuni di questi richiedono la licenza di caccia o pesca :



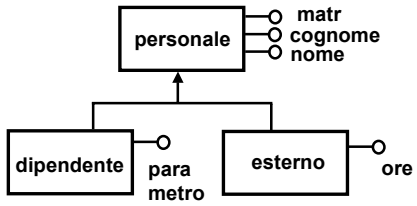
Proprieta' delle gerarchie

- ***t*** **sta per totale: ogni istanza dell'entità padre deve far parte di una delle entità figlie**
  - **nell'esempio il personale si divide (completamente) in esterni e dipendenti**
- ***p*** **sta per parziale: le istanze dell'entità padre possono far parte di una delle entità figlie**
  - **nell'esempio i cacciatori e pescatori sono un sottoinsieme dei cittadini**

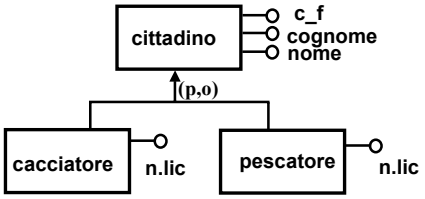
Proprieta' delle gerarchie

- ***e*** **sta per esclusiva: ogni istanza dell'entità padre non può far parte di più di una delle entità figlie**
  - **nell'esempio si esclude che una istanza di personale possa appartenere ad entrambe le sottoclassi**
- ***o*** **sta per overlpping: ogni istanza dell'entità padre può far parte di più entità figlie**
  - **nell'esempio un cittadono può essere al tempo stesso cacciatore e pescatore**

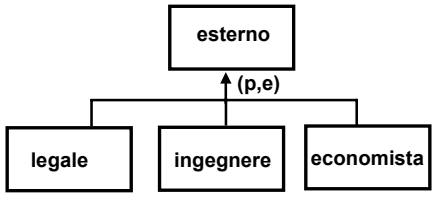
Default: (t,e)



Indicazione della proprietà

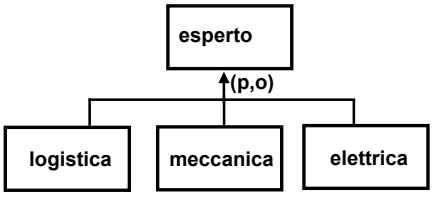


Un'ulteriore specializzazione



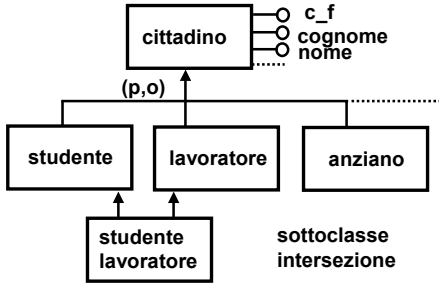
***p*: possono esistere esterni generici che non sono né legali, né ingegneri, né economisti ma non interessa stabilire una sottoclasse ad hoc**

Un'ulteriore specializzazione

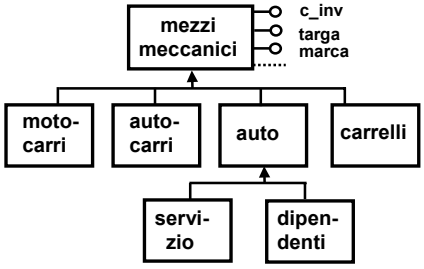


***p* : possono esistere esperti sia meccanici, sia elettrici, sia della logistica**
O: le tre qualifiche non si escludono

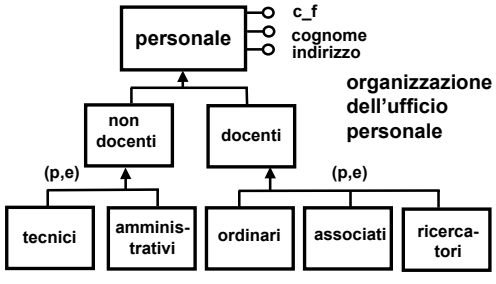
Esempio: un comune



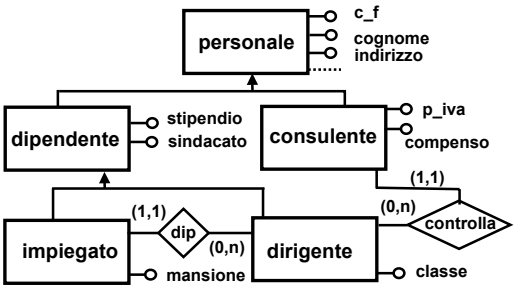
Esempio: parco mezzi meccanici



Esempio: università



Esempio: personale d'azienda



STRATEGIE DI PROGETTO

Strategie di progetto

- **Lo sviluppo dello schema si può eseguire seguendo due strategie fondamentali:**
  - **Top-Down**
  - **Bottom-Up**
- **Per quanto riguarda le strategie bottom-up, vedremo i casi:**
  - **"A macchia d'olio"**
  - **Mista**

Progetto top-down e bottom-up

- Top-down: si procede per raffinamenti a partire da una descrizione che comprende TUTTA la realtà di interesse
- Bottom-up: si disegnano separatamente aspettti della realtà e poi li si integra costruendo un unico schema

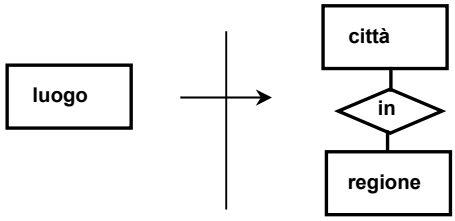
Confronto

- Top down: il progetto è più ordinato e razionale, ma più difficile (il progettista deve possedere una "visione d'assieme")
- Bottom-up: si possono prendere decisioni differenti nell'affrontare i sotto-problemi, che si tradurranno in "conflitti" (modelli diversi della stessa realtà).

Strategia top-down "pura"

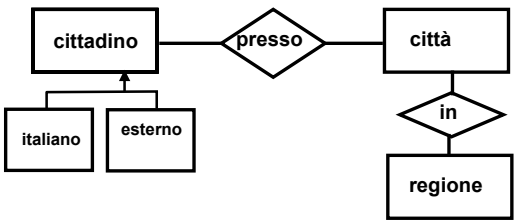
- **A partire dalle specifiche si costruisce uno schema iniziale**
- **Dallo schema iniziale si arriva per raffinamenti successivi a schemi intermedi e poi allo schema finale**
- **I raffinamenti prevedono l'uso di trasformazioni elementari (primitive) che operano sul singolo concetto per descriverlo con maggior dettaglio**

Trasformazione elementare top-down

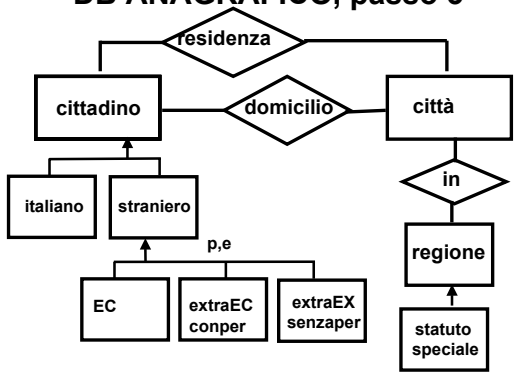


DB ANAGRAFICO, passo 1

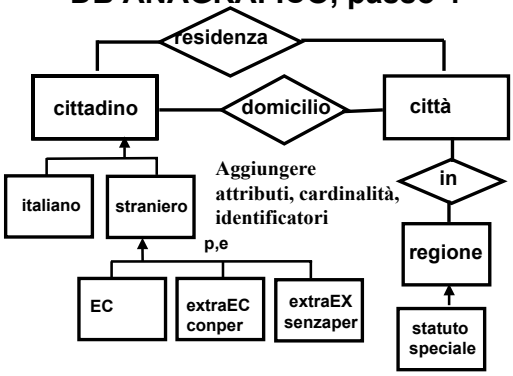
DB ANAGRAFICO, passo 2



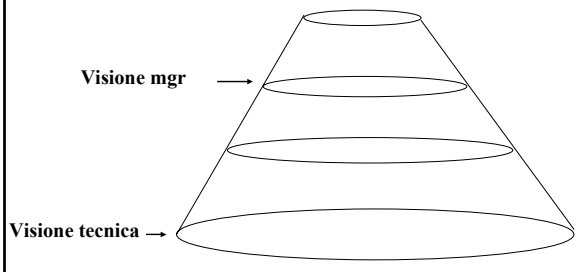
DB ANAGRAFICO, passo 3



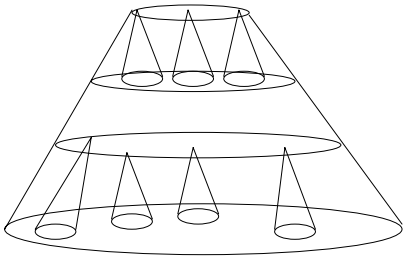
DB ANAGRAFICO, passo 4



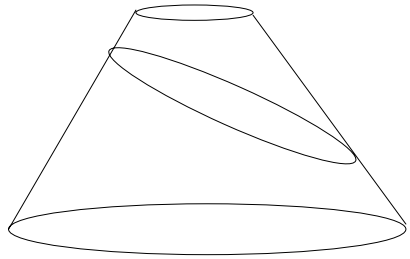
"Piani" del progetto



Progetto “equilibrato”



Progetto “squilibrato”



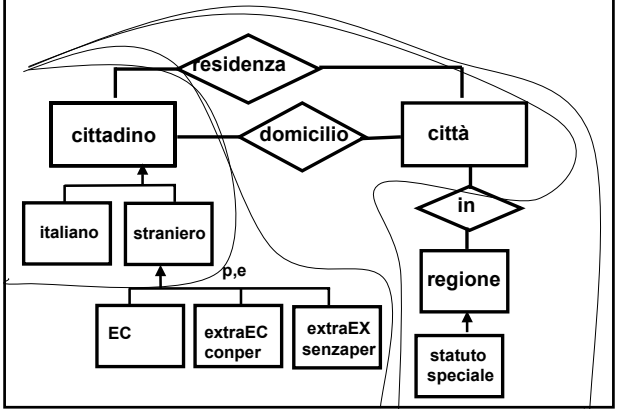
Valutazione di una strategia top down

- **vantaggi:**
  - **il progettista descrive inizialmente lo schema trascurando i dettagli**
  - **precisa lo schema gradualmente**
- **problema:**
  - **non va bene per applicazioni complesse perché è difficile avere una visione globale precisa iniziale di tutte le componenti del sistema**

Strategia “a macchia d’olio”

- **Le specifiche nascono progressivamente, affrontando i requisiti fino al massimo dettaglio e “avanzando” per sottoproblemi**
- **La tecnica è adatta a tradurre pian piano una descrizione testuale in un diagramma**

Strategia a macchia d’olio



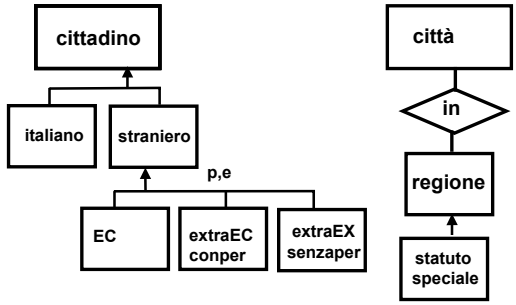
Valutazione della ìstrategia “a macchia d’olio”

- **La tecnica è adatta a tradurre pian piano una descrizione testuale in un diagramma**
- **Pur essendo bottom-up, il progettista analizza le specifiche in modo “stratificato” e le aggiunge progressivamente ad un unico schema, perciò i conflitti sono meno probabili**

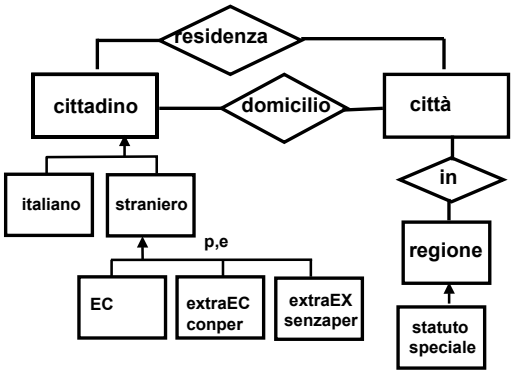
Strategia Mista

- Adatta di fronte a progetti ampi
- Si suddividono le specifiche in parti (ad esempio: le funzioni amministrazione, personale, marketing, vendita, produzione di una azienda)
- Si realizzano top-down le varie parti
- Si realizza (bottom-up) l'integrazione delle varie parti sviluppate

Sviluppo di due sottoschemi



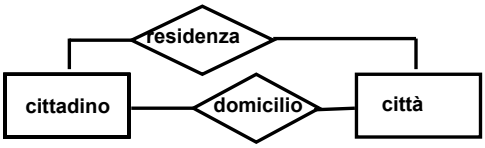
Integrazione



Valutazione della strategia mista

- **vantaggi:**
  - **diversi progettisti elaborano gli schemi parziali, il singolo progettista ha una visione più precisa del proprio settore**
- **problema:**
  - **conflitti e difficoltà di integrazione**
- **soluzione possibile:**
  - **sviluppare uno piccolo schema dei soli concetti principali (schema scheletro) in modo top-down e attenersi alle scelte presenti nello schema scheletro in tutti gli altri schemi.**

Schema Scheletro



Sintesi

- Un progettista esperto procede (inconsciamente) sia in modo top-down che in modo bottom-up
- Per affrontare gli esercizi, la tecnica a macchia d'olio viene usata spesso
- In ogni caso, è possibile (e molto conveniente) DOCUMENTARE un progetto in modo top-down a posteriori

QUALITA' DI UNO SCHEMA CONCETTUALE

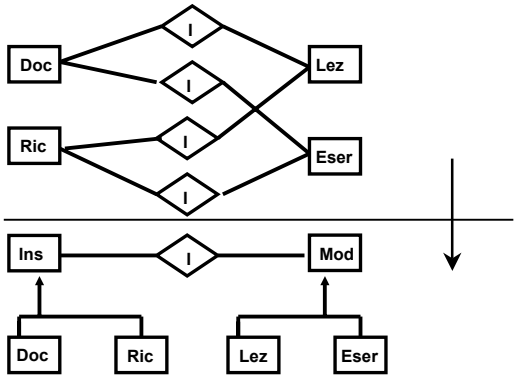
Qualità di uno schema concettuale

- Completezza
- Correttezza
- Leggibilità
- Minimalità
- Auto-Esplicatività

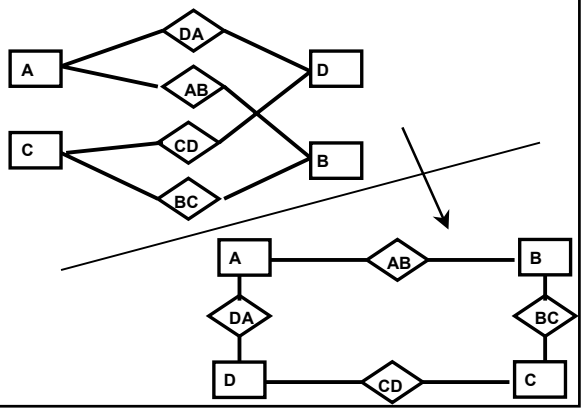
Completezza e correttezza

- Rappresentare in modo completo e corretto i requisiti
- Sono proprietà ovvie ma sulle quali c'è poco da aggiungere
  - Per la completezza: assicurarsi che i dati consentano di eseguire tutte le applicazioni
  - Per la correttezza: assicurarsi che sia possibile popolare la base di dati anche con informazione parzialmente incompleta durante fasi iniziali della sua evoluzione

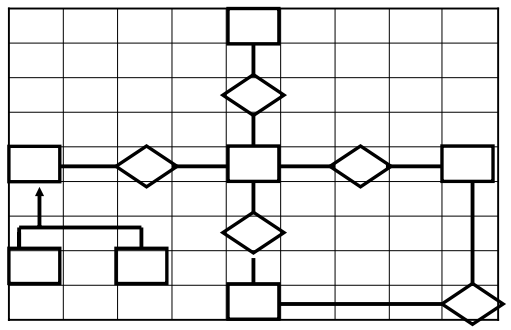
Leggibilità concettuale



Leggibilità grafica



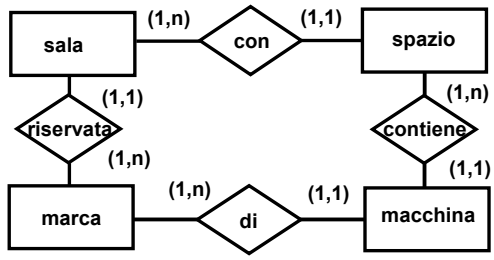
Disegnare in una griglia



Ridondanze negli schemi

- Una società gestisce delle sale di esposizione
- le sale di esposizione sono riservate a marche di macchine
- le sale comprendono spazi di esposizione
- gli spazi contengono macchine
- le macchine appartengono ad una certa marca

Ridondanze negli schemi

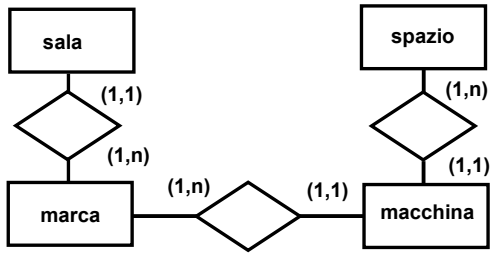


potrebbe esserci una ridondanza !

Ridondanze negli schemi

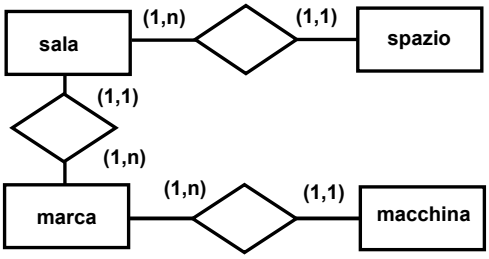
- il ciclo è ridondante se la sistemazione delle macchine negli spazi viene effettuata nel rispetto del vincolo che **una sala sia assegnata per intero ad una sola marca**
- proviamo ad eliminare le 4 associazioni a turno e verificare il rispetto delle specifiche

Eliminazione di: comprendere



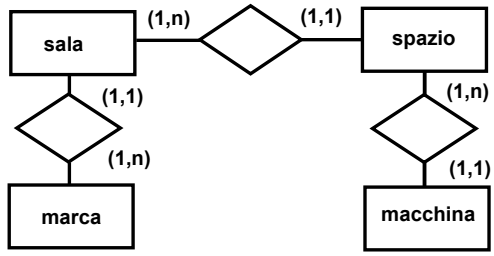
dato uno spazio non è possibile risalire alla sala che lo comprende

Eliminazione di: contenere

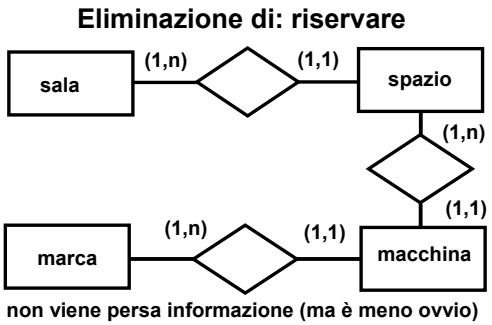


data una macchina non è possibile risalire allo spazio che la contiene

Eliminazione di: appartenere



non viene persa informazione

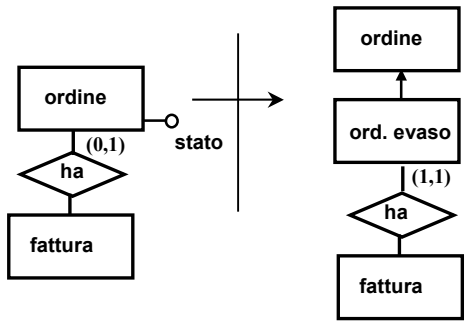


Discussione

- Ragionevole la prima eliminazione
- Piuttosto irragionevole la seconda eliminazione
- E in ogni caso: occorre conoscere bene il significato delle associazioni!

Auto-esplicatività

- Fare in modo che lo schema rappresenti esplicitamente il massimo di conoscenza sulla realtà



A questo punto....

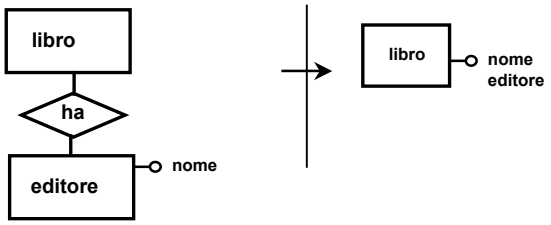
- Il progetto è stato condotto (o documentato) top-down
- Il progetto risponde ai requisiti di qualità

Ultimo passo: post-processing

Post-processing

- Verificare che:
  - Tutte le entità abbiano un identificatore
  - Tutte le associazioni abbiano cardinalità ben definite
  - Le entità siano significative (consentano di rappresentare più di una attributo o siano collegate ad altre entità tramite associazioni)
  - Le generalizzazioni siano utili (consentano di ereditare proprietà)

Esempio di entità inutile



Esempio di gerarchia inutile