

# **Manuale pratico di disegno e progettazione dei DataBase**

**Paolo Brunasti - 2010**

# Indice

INTRODUZIONE.....	5
Ciclo di vita di un sistema informativo.....	5
Ciclo di analisi e progettazione dei dati.....	5
Schemi e Istanze.....	5
Modelli.....	6
Motivi per l'uso dei modelli concettuali.....	6
Progettazione Concettuale.....	7
Raccolta dei requisiti.....	7
Documentazione di progetto.....	7
Business Rule.....	7
Scenari d'uso.....	8
Modello Entity-Relationship.....	8
Entità.....	8
Attributi.....	8
Relationship.....	8
Cardinalità.....	9
Identificatore.....	9
Generalizzazione.....	9
MODELLAZIONE DI UN DB.....	11
Formalismi.....	11
METODOLOGIA DI DISEGNO.....	12
Concetti di base del processo di disegno di un DB.....	12
Passi essenziali di disegno.....	12
Leggere il testo.....	12
Individuare le entità.....	13
Verificare se non si sono entità duplicate.....	13
Eventuale cambio dei nomi per semplicità.....	13
Individuare le associazioni tra le entità.....	13
Verificare se alcune associazioni hanno attributi.....	13
Calcolare la cardinalità delle associazioni.....	13
Trovare gli attributi espliciti delle entità.....	13
Definire le chiavi primarie delle entità.....	13
Identificazione di eventuali gerarchie.....	13
Stesura dello schema E-R.....	13
Ristrutturazione degli schemi E-R.....	13
Traduzione in modello relazionale.....	15
Entità.....	16
Associazioni molti a molti.....	16
Associazioni uno a molti.....	16
Associazioni uno a uno.....	16
Associazioni multiple.....	16
Identificatori esterni di entità.....	16
ESEMPI DI PROGETTAZIONE.....	17
Esempio 1 : Clienti.....	17
Testo di partenza.....	17
Individuare le entità.....	17
Verificare se non si sono entità duplicate.....	17
Individuare le associazioni tra le entità.....	17

Verificare se alcune associazioni hanno attributi.....	18
Calcolare la cardinalità delle associazioni.....	18
Trovare gli attributi espliciti delle entità.....	18
Definire le chiavi primarie delle entità.....	18
Identificazione di eventuali gerarchie.....	18
Stesura dello schema E-R.....	19
Esempio 2 : Sistema di Trasporti.....	20
Testo di partenza.....	20
Individuare le entità.....	20
Verificare se non si sono entità duplicate.....	21
Individuare le associazioni tra le entità.....	21
Verificare se alcune associazioni hanno attributi.....	21
Calcolare la cardinalità delle associazioni.....	21
Trovare gli attributi espliciti delle entità.....	21
Definire le chiavi primarie delle entità.....	22
Identificazione di eventuali gerarchie.....	22
Stesura dello schema E-R.....	23
Esempio 3 : Scuole elementari.....	24
Testo di partenza.....	24
Individuare le entità.....	24
Verificare se non si sono entità duplicate.....	24
Individuare le associazioni tra le entità.....	24
Verificare se alcune associazioni hanno attributi.....	25
Calcolare la cardinalità delle associazioni.....	25
Trovare gli attributi espliciti delle entità.....	25
Definire le chiavi primarie delle entità.....	26
Identificazione di eventuali gerarchie.....	26
Stesura dello schema E-R.....	27
Esempio 4 : Fatturazione.....	28
Testo di partenza.....	28
Individuare le entità principali.....	28
Verificare se non si sono entità duplicate.....	29
Individuare le associazioni tra le entità.....	29
Verificare se alcune associazioni hanno attributi.....	30
Calcolare la cardinalità delle associazioni.....	30
Trovare gli attributi espliciti delle entità.....	30
Definire le chiavi primarie delle entità.....	31
Identificazione di eventuali gerarchie.....	31
Stesura dello schema E-R.....	32
Esempio 5 : Elezioni amministrative.....	33
Testo di partenza.....	33
Individuare le entità.....	33
Verificare se non si sono entità duplicate.....	34
Eventuale cambio dei nomi per semplicità.....	34
Individuare le associazioni tra le entità.....	35
Verificare se alcune associazioni hanno attributi.....	36
Calcolare la cardinalità delle associazioni.....	36
Trovare gli attributi espliciti delle entità.....	36
Definire le chiavi primarie delle entità.....	37
Identificazione di eventuali gerarchie.....	37
Stesura dello schema E-R.....	38
Esempio X : xxx Scuole elementari.....	39
Testo di partenza.....	39

Individuare le entità.....	39
Verificare se non si sono entità duplicate.....	39
Individuare le associazioni tra le entità.....	39
Verificare se alcune associazioni hanno attributi.....	39
Calcolare la cardinalità delle associazioni.....	39
Trovare gli attributi espliciti delle entità.....	39
Definire le chiavi primarie delle entità.....	39
Identificazione di eventuali gerarchie.....	39
Stesura dello schema E-R.....	40
<b>PATTERN DI MODELLAZIONE.....</b>	<b>41</b>
Introduzione ai Pattern.....	41
Master-Detail.....	41

# INTRODUZIONE

## ***Ciclo di vita di un sistema informativo***

- **Idea**: identificazione di una necessità da soddisfare e del come soddisfarla
- **Studio di fattibilità**: verifica dell'effettiva realizzabilità dell'idea, valutazione costi e priorità
- **Raccolta e analisi dei requisiti**: raccolta dei desideri e delle necessità a cui va incontro il sistema e studio delle sue proprietà
- **Progettazione**: di dati e funzioni
- **Realizzazione**: codifica delle funzioni progettate e creazione dell'infrastruttura applicativa
- **Validazione e collaudo**: verifica sperimentale del funzionamento del sistema
- **Funzionamento**: vita operativa del sistema
- **Evoluzione**: trasformazione del sistema sulla base delle nuove esigenze (interne od esterne)

Centralità dei dati e necessità della loro modellazione in una fase precoce.

Il dato è più stabile delle funzioni (ma non immutabile)

## ***Ciclo di analisi e progettazione dei dati***

- **ANALISI (CHE COSA)**
  - raccolta dei requisiti
  - progettazione concettuale
    - schema concettuale
- **PROGETTAZIONE (COME)**
  - progettazione logica
    - schema logico
  - progettazione fisica
    - schema fisico

## **Schemi e Istanze**

**schema**: sostanzialmente invariante nel tempo, descrive la struttura (aspetto intensionale)

- nel modello relazionale, le intestazioni delle tabelle

**istanza**: i valori attuali, che cambiano anche molto rapidamente (aspetto estensionale)

- nel modello relazionale, il "corpo" di ciascuna tabella

## Modelli

**modelli concettuali:** permettono di rappresentare i dati in modo indipendente dal sistema

- cercano di descrivere i concetti del mondo reale
- sono utilizzati nelle fasi preliminari di progettazione

**modelli logici:** utilizzati per l'organizzazione dei dati da parte dei DBMS

- utilizzati dalle applicazioni
- indipendenti dalle strutture fisiche

## Motivi per l'uso dei modelli concettuali

Se partiamo a progettare direttamente dallo schema logico della base di dati di un'applicazione:

- da dove cominciamo?
- rischiamo di perderci subito nei dettagli
- dobbiamo pensare subito a come correlare le varie tabelle (chiavi etc.)
- ci sono grosse rigidità

Il modello concettuale ha:

- una rappresentazione grafica (più comunicativa)
- maggiore astrazione rispetto ai dettagli implementativi
- quindi più semplice
- quindi più comprensibile per l'utente/committente del sistema

La tecnica di modellazione concettuale più diffusa è quella “**Entity-Relationship**” (E-R)

## **Progettazione Concettuale**

### **Raccolta dei requisiti**

Dove si raccolgono i requisiti:

- dagli utenti del sistema
- nelle documentazioni relative al sistema
- da realizzazioni preesistenti (nello stesso contesto o in altri contesti)

Caratteristiche di una buona raccolta dei requisiti

- corretto livello di astrazione
- semplicità
- completamente esplicito (nulla di lasciato implicito)
- standardizzazione (dei termini e delle strutture)
- glossario

E' sempre e costantemente necessario verificare con l'utente la corretta comprensione da parte di entrambi e la consistenza e coerenza di tutto quanto descritto.

I requisiti non possono essere solo sui DATI, ma anche sulle OPERAZIONI da effettuare su tali dati, avendo quindi un approccio OLISTICO all'analisi.

### **Principio di indeterminatezza di Heisenberg (reciproco)**

*L'osservatore influenza il sistema osservato (ma anche il sistema osservato influenza l'osservatore).*

*Problema intrinseco della raccolta ed analisi dei requisiti: raccogliendo le informazioni se ne viene influenzati e quindi non si è più oggettivi, e interagendo con gli utenti non si è più neutrali.*

*Il rischio è conservare senza motivo caratteristiche dei sistemi e dei metodi esistenti senza metterli in discussione (e quindi compromettendo il nuovo sistema con eredità del vecchio), o essere prevenuti nei confronti di parti o utenti del sistema sulla base di quanto dettoci in precedenza.*

### **Documentazione di progetto**

Il modello E-R rappresenta in modo formale i concetti ma non li descrive, e non è adatto per la rappresentazione di vincoli complessi.

E' necessario integrare il modello E-R con una descrizione (formale o no) delle rimanenti parti del sistema.

### **Business Rule**

Le business rule relative all'analisi dei dati sono di tre tipi:

- descrizione di concetti – in forma testuale libera
- dichiarazione di vincoli d'integrità
  - asserzioni del tipo: <ente del modello> **deve / non deve** <proprietà o espressione>
- derivazione di una proprietà da altre esistenti
  - operazioni aritmetiche o logiche per dedurre proprietà o enti da altre informazioni presenti nel modello, del tipo: <risultato> **si ottiene** <espressione o formula>

Vi sono poi altre business rule che meglio si adattano alla modellazione dei processi aziendali, descrivendo appunto le regole aziendali che governano la trasformazione dei dati, ed hanno la forma:

**se** <condizione iniziale> **allora** <descrizione del risultato di un processo di trasformazione>

## Scenari d'uso

Uno strumento molto importante per l'analisi dei sistemi è la raccolta e la descrizione di scenari d'uso, cioè di precise sequenze di azioni ed eventi (come la sceneggiatura di un film) relativi a processi reali o da realizzare.

Sono utili soprattutto per la descrizione della parte dinamica del sistema, ma poiché i dati dovranno essere utilizzati all'interno di questo sistema dinamico, per ben progettare i dati devono essere noti anche i processi.

Permettono di far vedere (come in un film) all'utente come sarà il sistema, ma anche di formalizzare la descrizione dell'esistente per una più chiara comprensione da parte di progettisti ed analisti.

## Modello Entity-Relationship

Elementi e costrutti del modello E-R:

- **Entità** - rappresentazione di una categoria di oggetti del mondo reale che viene modellato
- **Attributo** – caratteristica dell'entità d'interesse nel modello (aggregabili in attributi composti)
- **Relazioni o associazioni** (Relationship) – collegamento tra diverse entità del modello
- **Cardinalità** – delle associazioni ma anche degli attributi
- **Identificatore** – insieme di attributi e/o associazioni che identificano un'entità
- **Generalizzazione** – classificazione gerarchica delle entità con ereditarietà delle caratteristiche (attributi, relazioni, etc..)

## Entità

Di solito si tratta di classi di oggetti (non singoli oggetti ma insiemi di oggetti con caratteristiche comuni) che possiedono un significato ed un'esistenza autonoma

Tutto dipende dal contesto applicativo, quella che può essere un'entità per un progetto potrebbe non esserlo in un altro (persone come entità in un sistema anagrafico, come valore numerico in un sistema geografico).

Ogni singola entità è indicata con un nome, espressivo all'intero del contesto applicativo.

Si seguono di solito delle opportune convenzioni, come quella di utilizzare nomi al singolare.

## Attributi

Sono in prima approssimazione i campi di una tabella.

Possono anche essere **COMPOSTI**.

## Relationship

Legame logico fra due o più entità, **rilevante** nell'applicazione di interesse

Il legame padre-figlio è fondamentale in un sistema per l'anagrafe civile, ininfluente in un sistema di gestione clienti.

Anche le associazioni hanno un nome, che sia possibilmente significativo e possibilmente non un verbo (“succedere”, “contiene”, ...) ma un sostantivo (“successione”, “componente”, ...).



E' possibile definire per le varie entità che sono collegate da una associazione degli specifici ruoli, che appunto dettagliano qualora sia necessario che parte gioca la singola entità nella associazione.

Tra due entità possono esservi più associazioni, ognuna con un suo particolare significato (ad esempio tra le entità "persona" e "città" possono esserci le associazioni "Residenza", "Nascita", "Domicilio",...).

Le associazioni possono essere tra più di due entità, ed in alcuni casi possono essere "opzionali" tra due o più entità.

Le associazioni possono essere anche ricorsive sulla stessa entità.

In questo caso di solito è necessario definire i ruoli all'interno della associazione che altrimenti potrebbe essere interpretata nel verso sbagliato (ad esempio per la associazione "genitore" sull'entità "persona" dev'essere indicato quale è la "madre" e quale il "figlio").

### **Promozione di una associazione**

Un attento esame delle associazioni a volte porta a "promuovere" una associazione e trasformarla in un'entità.

Ad esempio un esame visto in un primo momento come una associazione tra studente e corso può poi diventare per esigenze applicative un'entità autonoma con una associazione con studente ed una con corso.

Di solito questo avviene quando ad una associazione iniziano ad essere associati molti attributi.

### **Cardinalità**

Indica genericamente la numerosità di un ente.

Di due tipi:

- cardinalità di una associazione (o relazione)
- cardinalità di un attributo

La **cardinalità di una associazione** è una coppia di numeri interi che indica, per ogni "ramo" di una associazione (cioè per ogni entità che appare nella relazione), il numero minimo e massimo di volte che una associazione può esistere per una particolare occorrenza (oggetto) di un'entità.

Ad esempio la associazione ricorsiva "genitore" avrà una cardinalità 0-n per il ramo "madre" (cioè una persona può essere "madre" da 0 a n volte), mentre avrà una cardinalità 1-1 per il ramo "figlio" (cioè una persona deve sempre essere "figlia" una ed una sola volta).

### **Identificatore**

Insieme di attributi e/o di associazioni che identifica un'entità.

Gli identificatori sono i candidati a diventare le chiavi.

### **Generalizzazione**

Una serie di entità potrebbero avere delle caratteristiche in comune, come ad esempio un impiegato, uno studente, un professore: hanno tutte un nome, un cognome, una data di nascita, etc.

In questo caso si può pensare ad un'entità "Persona" che è la generalizzazione di tutte queste entità e definire che le varie impiegato, studente e professore derivano da questa entità generale, ereditando da essa tutte le caratteristiche e definendone eventualmente di nuove.



# MODELLAZIONE DI UN DB

## Formalismi

### Definizioni:

**entity** something about which data is collected, stored, and maintained

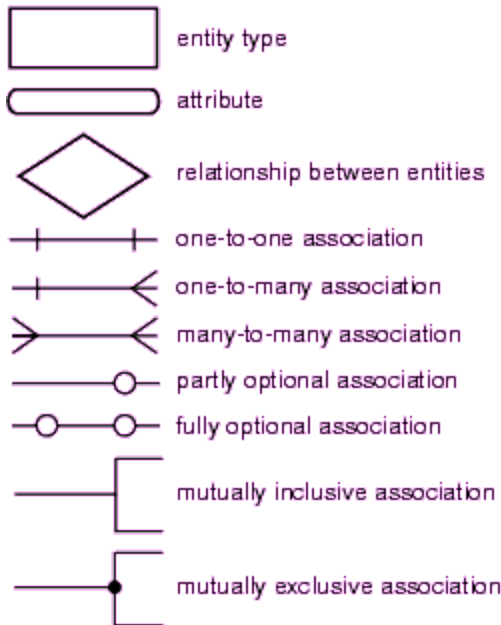
**attribute** a characteristic of an entity

**relationship** an association between entities

**entity type** a class of entities that have the same set of attributes

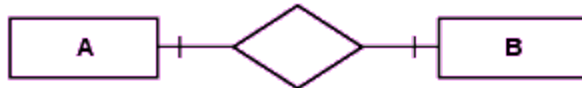
**record** an ordered set of attribute values that describe an instance of an entity type

### Symbols:

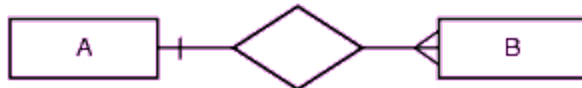


### Examples:

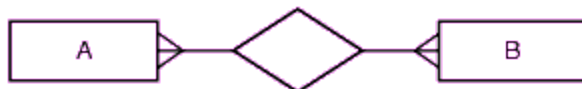
One A is associated with one B:



One A is associated with one or more B's:



One or more A's are associated with one or more B's:



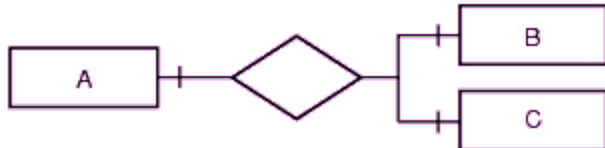
One A is associated with zero or one B:



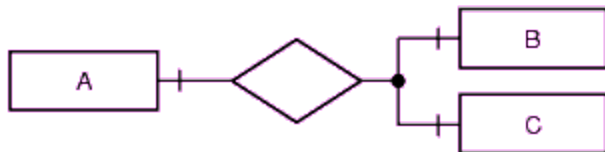
One A is associated with zero or more B's:



One A is associated with one B and one C:



One A is associated with one B or one C (but not both):



# METODOLOGIA DI DISEGNO

## ***Concetti di base del processo di disegno di un DB***

### ***Passi essenziali di disegno***

Il disegno di un DB può essere effettuato seguendo una serie di passi elementari, cioè “eseguendo” una specie di algoritmo.

Come già più volte detto non esiste un algoritmo unico e perfetto per il disegno di un DB a partire da delle specifiche testuali, sia perché esistono diversi metodi e strategia che possono essere vantaggiosi o problematici a seconda del problema specifico che si deve affrontare, ma anche per la diversa forma che può assumere la descrizione testuale di uno stesso problema, con maggiore o minore rigore formale.

Di seguito si fornisce quindi una traccia più che un algoritmo, che dev'essere seguita sempre con un forte spirito critico in modo da non farsi fuorviare del metodo rispetto al contesto specifico del problema.

1. Leggere il testo
2. Individuare le entità
3. Verificare se non si sono entità duplicate
  - Eventuale cambio dei nomi per semplicità
4. Individuare le associazioni tra le entità
5. Verificare se alcune associazioni hanno attributi
6. Calcolare la cardinalità delle associazioni
7. Trovare gli attributi espliciti delle entità
8. Definire le chiavi primarie delle entità
9. Stesura dello schema E-R

### **Leggere il testo**

Senza un'attenta lettura ed una comprensione complessiva del testo non si può sperare di creare un disegno corretto e soprattutto corrispondente al contenuto del testo.

Spesso una sola lettura, una singola passata, non è sufficiente per una piena comprensione, ed in alcuni casi può essere necessario rivolgersi a chi ha scritto il testo per comprendere meglio alcuni dettagli che risultano non chiari. Questo può essere dovuto a diversi fattori, che vanno dalla non condivisa base di conoscenza sul problema affrontato dal testo alla difficoltà intrinseca di tradurre in poche e semplici parole concetti che richiederebbero approfondite analisi. Un caso tipico è quello di concetti che lo scrittore dà per impliciti in quanto facenti parte del dominio delle conoscenze standard di un certo contesto operativo, ma che non sono poi di comune dominio e quindi immediatamente accessibili al lettore.

**Individuare le entità**

**Verificare se non si sono entità duplicate**

**Eventuale cambio dei nomi per semplicità**

**Individuare le associazioni tra le entità**

**Verificare se alcune associazioni hanno attributi**

**Calcolare la cardinalità delle associazioni**

**Trovare gli attributi espliciti delle entità**

**Definire le chiavi primarie delle entità**

**Identificazione di eventuali gerarchie**

**Stesura dello schema E-R**

### ***Ristrutturazione degli schemi E-R***

Serve per:

- facilitare la successiva traduzione dello schema E-R in uno schema relazionale.
- ottimizzazione delle prestazioni

Passi di ristrutturazione

- analisi delle ridondanze
- trasformazione delle generalizzazioni
- eliminazione attributi multi-valore
- accorpamento entità (associazioni 1-1)
- partizionamento verticale di entità (parte dei dati di un'entità non vengono quasi mai usati)
- partizionamento orizzontale di associazioni multiple
- identificatori principali

La ristrutturazione deve essere guidata da due obiettivi:

1. pulizia del disegno
2. ottimizzazione delle prestazioni

Questi due obiettivi spesso vanno di pari passo, ma a volte divergono, ed è quindi necessario trovare il giusto equilibrio tra “perfezione formale” del disegno e “sporca ottimizzazione” delle operazioni.

### ***Analisi delle ridondanze***

Le ridondanze creano problemi di allineamento dei dati, ma in alcuni casi possono velocizzare le interrogazioni.

Vantaggi:

- interrogazioni più semplici

Svantaggi:

- complicazione ed appesantimento degli aggiornamenti e **rischio di inconsistenza**
- maggiore occupazione di spazio

Le ridondanze possono essere dovute:

- attributi ripetuti tra entità
- attributi i cui valori possono essere altresì derivati da altri attributi della stessa entità
- attributi i cui valori possono essere derivati da attributi di altre entità
- attributi i cui valori possono essere derivati da conteggi su entità
- associazioni derivabili da altre associazioni

### ***Trasformazione delle generalizzazioni***

Le generalizzazioni possono essere trasformate secondo tre linee principali, in base alle tipologie di partizionamento delle informazioni tra entità generale ed entità di dettaglio, sia dal punto di vista della completezza che dell'uso che ne viene fatto.

Le tre linee guida sono:

1. un'unica tabella che accorpa entità generale e tutte le tipologie di dettaglio (minimo comune multiplo, 1 sola tabella) – necessario metodo per identificare il tipo
2. replica della struttura del generale nelle varie entità di dettaglio ( n tabelle)
3. entità separate per parte generale e per i vari dettagli (1 + n tabelle)

Ogni linea presenta vantaggi e svantaggi che vanno pesate sulla base dell'uso che si fa dei dati.

1. conviene se gli accessi al generale ed al dettaglio sono contestuali

2. conviene se gli accessi ai dettagli sono distinti per tipologia

3. conviene se gli accessi ai dettagli sono separati dagli accessi alle informazioni generali

### ***Eliminazione attributi multi-valore***

Gli attributi multivalore possono essere mappati con una tabella di dettaglio con una associazione uno a molti con quella di origine.

Intrinsecamente i campi multivalore hanno dei limiti dal punto di vista della numerosità massima gestibile e dell'accesso al singolo valore, problemi che possono essere superati trasformando appunto in una tabella di dettaglio che mantiene una dipendenza con l'entità principale.

### ***Accorpamento entità (associazioni 1-1) - Partizionamento verticale di entità***

Ogni volta che si presenta una associazione 1-1 tra due entità si deve valutare se tale associazione non nasconda in realtà un'identità tra le due entità coinvolte, cioè che le due entità modellate rappresentino aspetti diversi della stessa entità reale.

Dev'essere quindi valutata attentamente l'opportunità di accorpate in un'unica entità gli attributi delle due entità di partenza, tenendo presente che a volte considerazioni di carattere prestazionale possono spingere a non effettuare tale operazione.

Nel senso opposto le varie entità devono essere analizzate in funzione delle interrogazioni svolte su di esse in modo da ottimizzarne l'accesso.

Infatti se una parte dei dati non viene quasi mai acceduta può essere “parcheggiata” in una entità associata che pur mantenendo le informazioni non appesantisce le interrogazioni “abitudinali”.

Oppure se alcune interrogazioni si focalizzano su una parte di dati ed altre su una parte diversa, queste parti possono essere focalizzate “specializzando” due sotto entità in funzione delle interrogazioni.

### ***Partizionamento orizzontale di associazioni multiple***

Una associazione 1-n o n-m può avere caratteristiche di accesso particolari, cioè il legame può essere prioritario su una parte di questa associazione, soprattutto quando tale associazione descrive la storia di un'entità o di una associazione che varia nel tempo. Ad esempio una associazione tra un'automobile e tutti i suoi proprietari avrà una focalizzazione sul proprietario attuale, essendo gli altri temporalmente superati.

In tal caso si può prendere in considerazione l'opportunità di dividere la relazione originaria in due relazioni che rappresentano una la caratteristica prioritaria (ad esempio la situazione corrente) e l'altra tutte le caratteristiche secondarie (ad esempio lo storico di tutte le relazioni).

### ***Identificatori principali***

Un passo fondamentale è l'analisi delle entità per l'individuazione degli identificatori principali (o chiavi) delle entità stesse.

I criteri di base per l'individuazione degli attributi adatti sono:

- assenza di opzionalità (cioè not null)
- minimalità (dannoso aggiungere attributi superflui)
- valutazione dell'utilizzo relativamente alle varie operazioni

Se nessun insieme di attributi soddisfa queste caratteristiche si può ricorrere alla creazione di nuovi attributi ad hoc: i **codici identificativi**.

Tali codici assumono valori generati appositamente, di solito attraverso **SEQUENCE**

### ***Traduzione in modello relazionale***

Passi di traduzione

- entità
- associazioni molti a molti
- associazioni uno a molti
- associazioni uno a uno
- associazioni multiple
- identificatori esterni di entità

## Entità

Le entità si mappano direttamente in relazioni che hanno come attributi gli stessi dell'entità originante.

### Associazioni molti a molti

Si trasformano in una relazione che ha come attributi le chiavi delle due entità che sono collegate da tale associazione, oltre agli attributi della associazione stessa.

L'unica limitazione è nella cardinalità della associazione poiché i valori minimi e massimi possono essere imposti solo attraverso complicati costrutti.

### Associazioni uno a molti

L'associazione viene espressa come attributo nella relazione che partecipa con cardinalità maggiore, che referenziano le chiavi dell'altra entità, più gli eventuali attributi dell'associazione stessa.

Infatti date le due entità coinvolte A e B con cardinalità 1 ed n, l'associazione esisterà al massimo per ogni occorrenza dell'entità B, cioè per ogni tupla della relazione che mappa B.

Se l'associazione ha cardinalità minima 0, cioè non è obbligatoria, gli attributi di chiave e dell'associazione saranno “nullable”.

### Associazioni uno a uno

Tale tipo di associazione può essere:

- fusa in una delle due entità (relazioni) coinvolte
- portare a fondere completamente anche le due entità dell'associazione

La scelta dipende dalle caratteristiche applicative di accesso ai dati.

### Associazioni multiple

Le associazioni multiple vengono tradotte in modo simile a quanto fatto per le associazioni più semplici, con l'accortezza di identificare quali sono le relazioni che meglio si adattano ad essere usate come punto di aggregazione per le chiavi delle altre relazioni.

### Identificatori esterni di entità

L'identificatore della relazione verrà definito aggiungendo anche l'identificatore della associazione esterna



# ESEMPI DI PROGETTAZIONE

## **Esempio 1 : Clienti**

### **Testo di partenza**

Una società crea un archivio per gestire i propri clienti.

Ogni cliente ha una sola ragione sociale e una sola partita IVA, ma può avere più sedi.

Di ogni cliente interessa poi registrare i vari contatti, corrispondenti alle diverse persone della ditta cliente con le quali vengono intrattenute relazioni.

La società è strutturata in diversi uffici, ognuno dei quali gestisce una serie di clienti.

Lo stesso cliente può essere gestito da uno o più uffici.

### **Individuare le entità**

Una *società* crea un *archivio* per gestire i propri *clienti*.

Ogni *cliente* ha una sola ragione sociale e una sola partita IVA, ma può avere più *sedi*.

Di ogni *cliente* interessa poi registrare i vari *contatti*, corrispondenti alle diverse *persone* della ditta cliente con le quali vengono intrattenute relazioni.

La *società* è strutturata in diversi *uffici*, ognuno dei quali gestisce una serie di *clienti*.

Lo stesso *cliente* può essere gestito da uno o più *uffici*.

Abbiamo quindi identificato le seguenti possibili entità:

- Società
- Archivio
- Clienti
- Sedi (dei clienti)
- Contatti
- Persone
- Uffici (della società)

### **Verificare se non si sono entità duplicate**

Contatti – Persone

Decidiamo di utilizzare il termine **Contatto**

Società – Archivio

Non si tratta della stessa entità dal punto di vista del mondo reale, ma dal punto di vista del sistema informativo possono essere considerate coincidenti, utilizziamo il termine **Società**

### **Individuare le associazioni tra le entità**

Una *società* crea un *archivio per gestire* i propri *clienti*.

Ogni *cliente* ha una sola ragione sociale e una sola partita IVA, ma può avere più sedi.

Di ogni *cliente* interessa poi registrare i vari *contatti*, corrispondenti alle diverse *persone* della ditta cliente con le quali vengono intrattenute relazioni.

La *società* è strutturata in diversi uffici, ognuno dei quali gestisce una serie di *clienti*.

Lo stesso *cliente* può essere gestito da uno o più *uffici*.

Possiamo quindi identificare le seguenti associazioni:

- *società per gestire clienti.*
- *cliente può avere più sedi.*
- *cliente registrare contatti*
- *società è strutturata in diversi uffici*
- *uffici gestisce clienti.*
- *cliente gestito da uffici*

## Verificare se alcune associazioni hanno attributi

In questo caso non ci sono attributi legati ad associazioni.

## Calcolare la cardinalità delle associazioni

Le due ipotesi di associazioni (*uffici gestisce clienti*) e (*cliente gestito da uffici*) coinvolgono le stesse entità e rappresentano i due versi della stessa associazione.

- *società per gestire clienti* - 1 ad N (opzionale per società)
- *cliente può avere più sedi* - 1 ad N
- *cliente registrare contatti* - 1 ad N
- *società è strutturata in diversi uffici* - 1 ad N
- (*uffici gestisce clienti*) & (*cliente gestito da uffici*) - N ad N

## Trovare gli attributi espliciti delle entità

Una *società* crea un *archivio* per gestire i propri *clienti*.

Ogni *cliente* ha una sola *ragione sociale* e una sola *partita IVA*, ma può avere più *sedi*.

Di ogni *cliente* interessa poi registrare i vari *contatti*, corrispondenti alle diverse *persone* della ditta cliente con le quali vengono intrattenute relazioni.

La *società* è strutturata in diversi *uffici*, ognuno dei quali gestisce una serie di *clienti*.

Lo stesso *cliente* può essere gestito da uno o più *uffici*.

Nell'esempio si hanno quindi solo i seguenti attributi tutti relativi all'entità CLIENTE:

- ragione sociale
- partita IVA

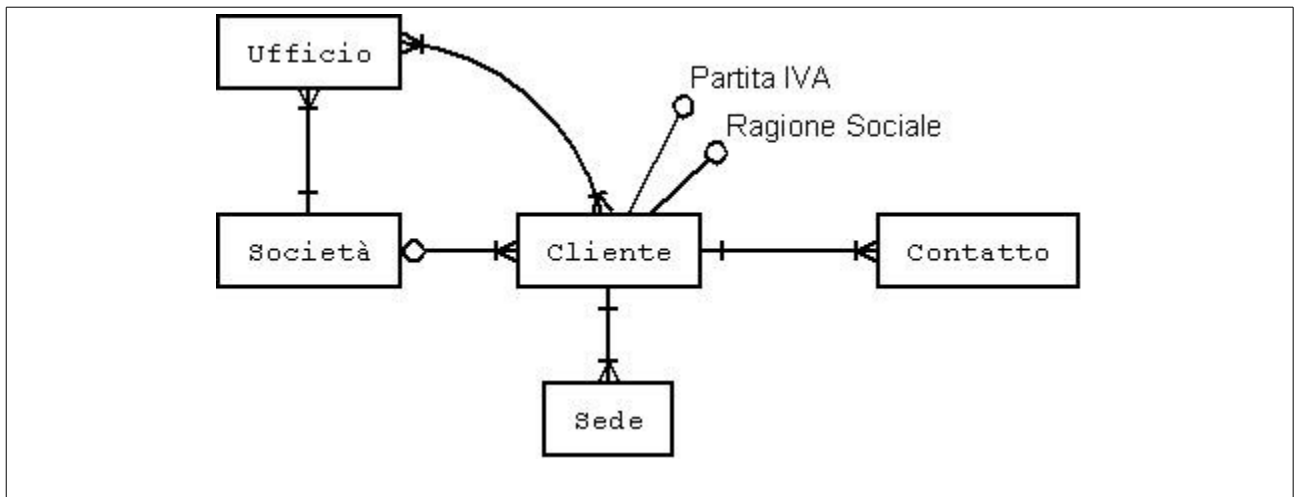
## Definire le chiavi primarie delle entità

Nel testo non sono indicate in modo esplicito le chiavi primarie delle entità.

## Identificazione di eventuali gerarchie

In questo esempio non ci sono gerarchie identificabili.

## Stesura dello schema E-R



## **Esempio 2 : Sistema di Trasporti**

### **Testo di partenza**

Si vuole realizzare un DB per gestire i trasporti pubblici di una città.

Si hanno una serie di linee di trasporto, ogni linea è caratterizzata da fermate raggruppate in fasce (a seconda della distanza dal centro cittadino) ed in settori (in modo che nelle fasce extraurbane il passaggio da un settore all'altro comporti cambi di tariffa).

Ogni linea è caratterizzata da una serie di "Tabelle", ognuna corrispondente alla tabella di marcia di una particolare corsa, caratterizzata dall'orario in cui un dato veicolo dovrà passare da una particolare fermata della linea.

Le linee sono suddivise in due categorie principali, quelle di superficie (per le quali le fermate sono identificate da un indirizzo copreso il numero civico davanti al quale viene effettuata la fermata) e quelle metropolitane (le cui fermate hanno invece solo un nome ed un indirizzo generico non riportante il numero civico, ma spesso il nome di due strade o di una piazza).

Inoltre le linee di superficie sono di diverso tipo, sulla base del mezzo che viene utilizzato (tram, bus, filobus, ...).

Si vuole anche memorizzare il sistema di tariffazione in modo semplificato considerando solo i cambi di fascia, quindi solo la tariffa corrispondente ad i vari possibili cambi di fascia o relativa al movimento circoscritto all'interno della fascia stessa.

### **Individuare le entità**

Si vuole realizzare un DB per gestire i trasporti pubblici di una città.

Si hanno una serie di **linee di trasporto**, ogni linea è caratterizzata da **fermate** raggruppate in **fasce** (a seconda della distanza dal centro cittadino) ed in **settori** (in modo che nelle **fasce** extraurbane il passaggio da un settore all'altro comporti cambi di tariffa).

Ogni **linea** è caratterizzata da una serie di "**Tabelle**", ognuna corrispondente alla tabella di marcia di una particolare **corsa**, caratterizzata dall'**orario** in cui un dato **veicolo** dovrà passare da una particolare **fermata** della **linea**.

Le **linee** sono suddivise in due **categorie** principali, quelle di superficie (per le quali le **fermate** sono identificate da un indirizzo compreso il numero civico davanti al quale viene effettuata la fermata) e quelle metropolitane (le cui **fermate** hanno invece solo un nome ed un indirizzo generico non riportante il numero civico, ma spesso il nome di due strade o di una piazza).

Inoltre le **linee** di superficie sono di diverso **tipo**, sulla base del mezzo che viene utilizzato (tram, bus, filobus, ...).

Abbiamo quindi identificato le seguenti possibili entità:

- linee
- fermate
- fasce
- settori
- tabelle
- corse
- orario
- veicolo
- categorie
- linee di superficie
- tipo (linea di superficie)

## Verificare se non si sono entità duplicate

Tabelle - Corse

utilizziamo il termine **Corse**

## Individuare le associazioni tra le entità

Si vuole realizzare un DB per gestire i trasporti pubblici di una città.

Si hanno una serie di **linee di trasporto**, ogni **linea** è caratterizzata da **fermate raggruppate in fasce** (a seconda della distanza dal centro cittadino) ed in settori (in modo che nelle fasce extraurbane il passaggio da un settore all'altro comporti cambi di tariffa).

Ogni **linea** è caratterizzata da una serie di “**Tabelle**”, ognuna corrispondente alla tabella di marcia di una particolare **corsa**, caratterizzata dall'orario in cui un dato veicolo dovrà passare da una particolare **fermata** della **linea**.

Le **linee** sono suddivise in due **categorie** principali, quelle di superficie (per le quali le **fermate** sono identificate da un indirizzo compreso il numero civico davanti al quale viene effettuata la fermata) e quelle metropolitane (le cui **fermate** hanno invece solo un nome ed un indirizzo generico non riportante il numero civico, ma spesso il nome di due strade o di una piazza).

Inoltre le **linee di superficie** sono di diverso **tipo**, sulla base del mezzo che viene utilizzato (tram, bus, filobus, ...).

Possiamo quindi identificare le seguenti associazioni:

- **linea** è caratterizzata da **fermate**
- **fermate** raggruppate in **fasce**
- **fermate** raggruppate in **settori**
- **linea** è caratterizzata da “**Tabelle**” (**corse**)
- **corsa**, caratterizzata dall'orario
- **corsa** .... in cui un dato **veicolo**
- **orario** in cui dovrà passare da **fermata**
- **linee** sono suddivise in **categorie** (*questa sarà una gerarchia*)
- **linee di superficie** sono di **tipo**

## Verificare se alcune associazioni hanno attributi

In questo caso non ci sono attributi legati a associazioni.

## Calcolare la cardinalità delle associazioni

- **linea** è caratterizzata da **fermate** - 1 ad N
- **fermate** raggruppate in **fasce** - N ad 1
- **fermate** raggruppate in **settori** - N ad 1
- **linea** è caratterizzata da “**Tabelle**” (**corse**) - 1 ad N
- **corsa**, caratterizzata dall'orario - 1 ad N
- **corsa** .... in cui un dato **veicolo** - N ad M
- **orario** in cui dovrà passare da **fermata** - N ad 1
- **linee** sono suddivise in **categorie** - N ad 1
- **linee di superficie** sono di **tipo** - N ad 1

## Trovare gli attributi espliciti delle entità

Si vuole realizzare un DB per gestire i trasporti pubblici di una città.

Si hanno una serie di **linee di trasporto**, ogni linea è caratterizzata da **fermate raggruppate in fasce**

(a seconda della *distanza* dal centro cittadino) ed in **settori** (in modo che nelle **fasce** extraurbane il passaggio da un settore all'altro comporti cambi di tariffa).

Ogni **linea** è caratterizzata da una serie di “**Tabelle**”, ognuna corrispondente alla tabella di marcia di una particolare **corsa**, caratterizzata dall'**orario** in cui un dato **veicolo** dovrà passare da una particolare **fermata** della **linea**.

Le **linee** sono suddivise in due **categorie** principali, quelle di superficie (per le quali le **fermate** sono identificate da un *indirizzo compreso il numero civico* davanti al quale viene effettuata la fermata) e quelle metropolitane (le cui **fermate** hanno invece solo *un nome* ed *un indirizzo generico non riportante il numero civico, ma spesso il nome di due strade o di una piazza*).

Inoltre le **linee** di superficie sono di diverso **tipo**, sulla base del mezzo che viene utilizzato (tram, bus, filobus, ...).

Nell'esempio si hanno quindi i seguenti attributi relativi all'entità FASCIA:

- distanza (dal centro)

Attributi relativi all'entità FERMATA:

- nome (per le fermate della metropolitana)
- indirizzo
- secondo indirizzo (eventuale, per le fermate della metropolitana)

## Definire le chiavi primarie delle entità

Chiave primaria relativi all'entità FERMATA:

- nome (per le fermate della metropolitana)
- indirizzo(per le fermate di superficie)

## Identificazione di eventuali gerarchie

In questo esempio possono essere identificate le seguenti gerarchie:

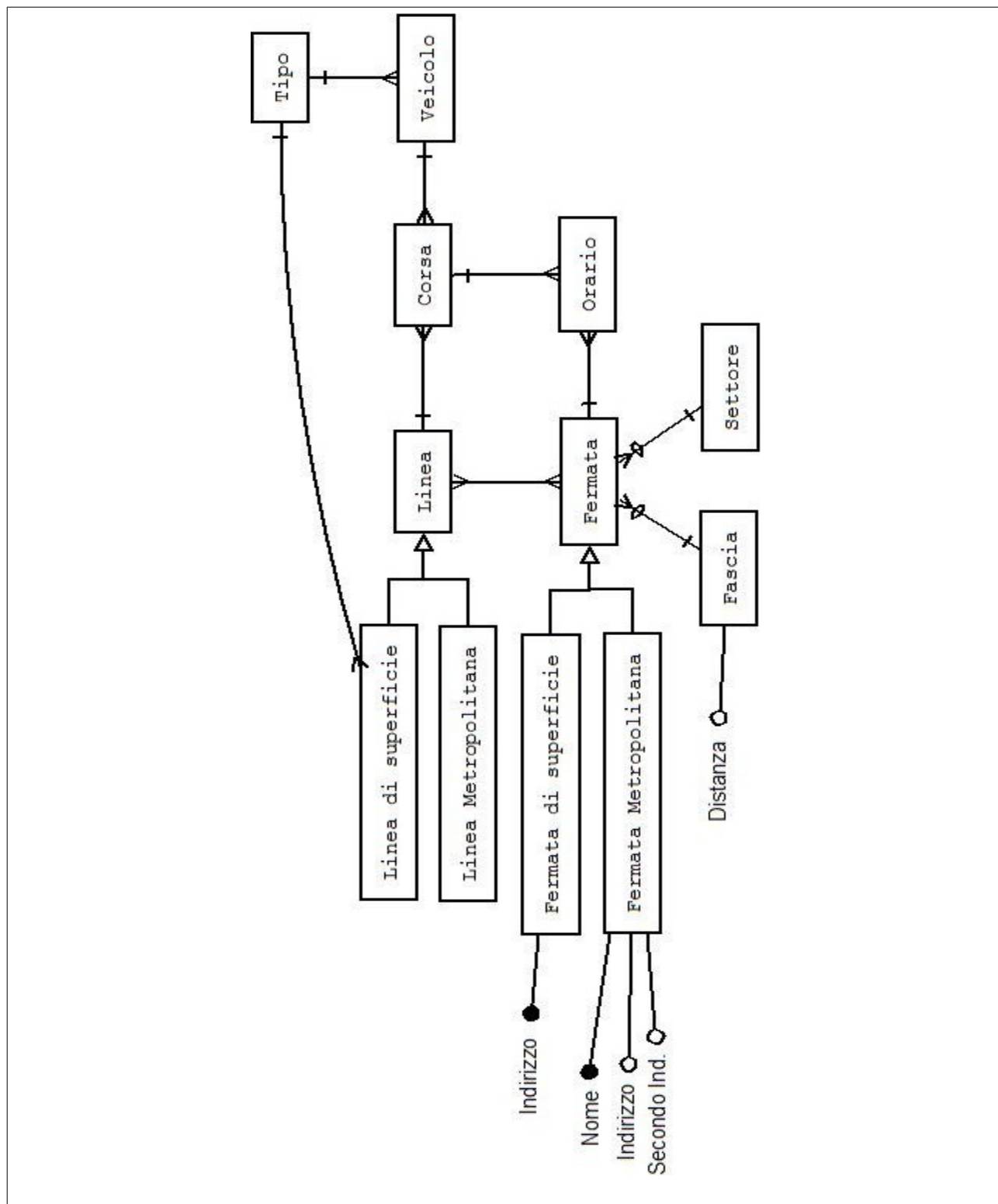
Linea

- Superficie
- Metropolitana

Fermate

- Superficie
- Metropolitana

# Stesura dello schema E-R



## **Esempio 3 : Scuole elementari**

### **Testo di partenza**

Si vuole realizzare un DB per gestire una serie di scuole elementari (limitatamente al singolo anno scolastico).

Ogni scuola ha un indirizzo, ed è composta da una serie di classi, ognuna a sua volta composta da alunni.

Ogni alunno ha nome, cognome, data di nascita ed un indirizzo di residenza.

Vi sono poi i maestri, che hanno a loro volta nome, cognome, data di nascita ed un indirizzo di residenza, e possono essere assegnati ad una sola classe, nel qual caso sono detti “monoclasse”, od operare su più classi durante l’anno (ad esempio i maestri di ginnastica o di lingua), nel qual caso sono detti “trasversali”. Tutti i maestri indipendentemente dalla tipologia operano in una ed una sola scuola. Ogni maestro “monoclasse” può insegnare più materie per quella classe, mentre i maestri “trasversali” insegnano una sola materia su più classi. Per ogni scuola viene identificato, tra i suoi maestri, uno ed un solo “responsabile didattico”.

### **Individuare le entità**

Si vuole realizzare un DB per gestire una serie di *scuole elementari* (limitatamente al singolo anno scolastico).

Ogni *scuola* ha un indirizzo, ed è composta da una serie di *classi*, ognuna a sua volta composta da *alunni*.

Ogni *alunno* ha nome, cognome, data di nascita ed un indirizzo di residenza.

Vi sono poi i *maestri*, che hanno a loro volta nome, cognome, data di nascita ed un indirizzo di residenza, e possono essere assegnati ad una sola *classe*, nel qual caso sono detti “*monoclasse*”, od operare su più *classi* durante l’anno (ad esempio i maestri di ginnastica o di lingua), nel qual caso sono detti “*trasversali*”. Tutti i *maestri* indipendentemente dalla tipologia operano in una ed una sola *scuola*. Ogni *maestro* “*monoclasse*” può insegnare più *materie* per quella *classe*, mentre i *maestri* “*trasversali*” insegnano una sola *materia* su più *classi*. Per ogni *scuola* viene identificato, tra i suoi *maestri*, uno ed un solo “*responsabile didattico*”.

Abbiamo quindi identificato le seguenti possibili entità:

- scuole
- classi
- alunni
- maestri “monoclasse”
- maestri “trasversali”
- materie
- responsabile didattico

### **Verificare se non si sono entità duplicate**

Maestro – Responsabile Didattico

Il responsabile didattico è uno tra i maestri, quindi non è un'entità a parte, ma un qualche attributo di un maestro o una particolare associazione tra esso e una scuola

### **Individuare le associazioni tra le entità**

Si vuole realizzare un DB per gestire una serie di *scuole elementari* (limitatamente al singolo anno scolastico).

Ogni *scuola* ha un indirizzo, ed è composta da una serie di *classi*, ognuna a sua volta composta da *alunni*.



Ogni **alunno** ha nome, cognome, data di nascita ed un indirizzo di residenza.

Vi sono poi i **maestri**, che hanno a loro volta nome, cognome, data di nascita ed un indirizzo di residenza, e possono essere assegnati ad una sola classe, nel qual caso sono detti "**monoclasse**", od operare su più classi durante l'anno (ad esempio i maestri di ginnastica o di lingua), nel qual caso sono detti "**trasversali**". Tutti i **maestri** indipendentemente dalla tipologia operano in una ed una sola scuola. Ogni **maestro "monoclasse"** può insegnare più materie per quella **classe**, mentre i **maestri "trasversali"** insegnano una sola materia su più classi. Per ogni **scuola** viene identificato, tra i suoi **maestri**, uno ed un solo "responsabile didattico".

Possiamo quindi identificare le seguenti associazioni:

- **scuola** è composta da classi
- **classe** composta da alunni.
- **maestri "monoclasse"** possono essere assegnati ad una sola classe
- **maestri "trasversali"** operare su più classi
- **maestri** operano in una ed una sola scuola
- **maestro "monoclasse"** può insegnare più materie
- **maestri "trasversali"** insegnano una sola materia
- Per ogni **scuola** tra i suoi **maestri**, uno ed un solo "responsabile didattico".

## Verificare se alcune associazioni hanno attributi

In questo caso non ci sono attributi legati a associazioni.

## Calcolare la cardinalità delle associazioni

- **scuola** è composta da classi - 1 ad N
- **classe** composta da alunni - 1 ad N
- **maestri "monoclasse"** possono essere assegnati ad una sola classe - N ad 1
- **maestri "trasversali"** operare su più classi - N ad N
- **maestri** operano in una ed una sola scuola - N ad 1
- **maestro "monoclasse"** può insegnare più materie - N ad N
- **maestri "trasversali"** insegnano una sola materia - N ad 1
- **Per ogni scuola** tra i suoi **maestri**, uno ed un solo "responsabile didattico" - 1 ad 1

## Trovare gli attributi espliciti delle entità

Si vuole realizzare un DB per gestire una serie di **scuole elementari** (limitatamente al singolo anno scolastico).

Ogni **scuola** ha un indirizzo, ed è composta da una serie di **classi**, ognuna a sua volta composta da **alunni**.

Ogni **alunno** ha nome, cognome, data di nascita ed un indirizzo di residenza.

Vi sono poi i **maestri**, che hanno a loro volta nome, cognome, data di nascita ed un indirizzo di residenza, e possono essere assegnati ad una sola **classe**, nel qual caso sono detti "**monoclasse**", od operare su più **classi** durante l'anno (ad esempio i maestri di ginnastica o di lingua), nel qual caso sono detti "**trasversali**". Tutti i **maestri** indipendentemente dalla tipologia operano in una ed una sola **scuola**. Ogni **maestro "monoclasse"** può insegnare più **materie** per quella **classe**, mentre i **maestri "trasversali"** insegnano una sola **materia** su più **classi**. Per ogni **scuola** viene identificato, tra i suoi **maestri**, uno ed un solo "**responsabile didattico**".

Abbiamo quindi i seguenti attributi:

Scuola

- indirizzo

#### Alunno

- nome
- cognome
- data di nascita
- indirizzo di residenza

#### Maestro

- nome
- cognome
- data di nascita
- indirizzo di residenza

### **Definire le chiavi primarie delle entità**

Nel testo non sono indicate in modo esplicito le chiavi primarie delle entità.

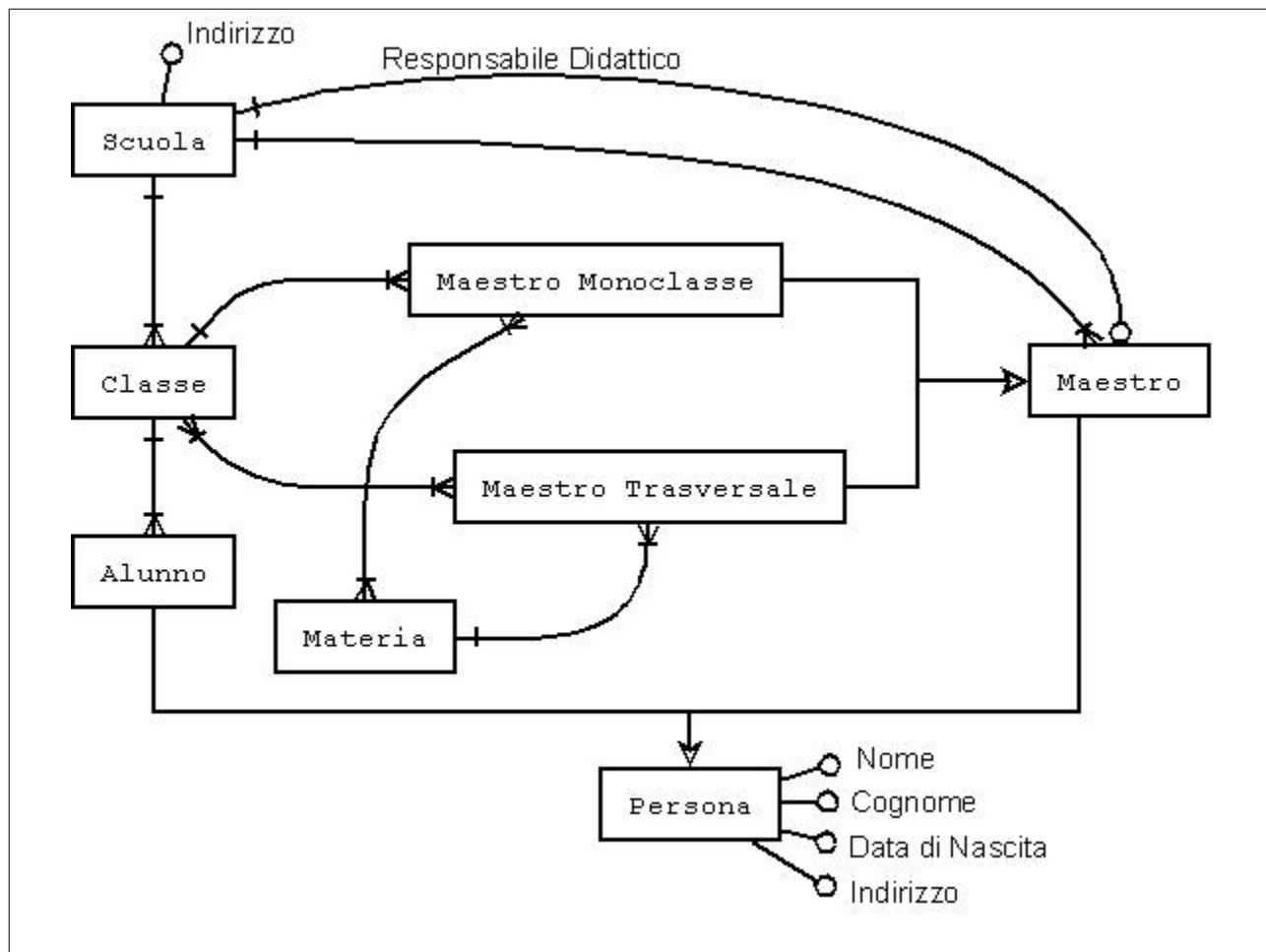
### **Identificazione di eventuali gerarchie**

In questo esempio possono essere identificate le seguenti gerarchie:

#### Persona

- Allievo
- Maestro
  - “Monoclasse”
  - “Trasversale”

## Stesura dello schema E-R



## Esempio 4 : Fatturazione

### Testo di partenza

Si vuole realizzare un DB per gestire la fatturazione di una società.

L'attività principale dell'azienda è gestire pratiche tecnico-legali per i propri clienti.

Una singola pratica è associata a più clienti ed ha un numero di riferimento interno che la identifica univocamente, una data apertura ed uno stato operativo.

Ogni cliente può avere più indirizzi associati, ma se sono più di uno allora uno ed uno solo tra questi sarà indicato come indirizzo di fatturazione.

Il computo degli importi da fatturare è fatto a partire da voci di spesa legate alle singole pratiche, dove ogni voce di spesa ha associato un operatore che ha svolto l'attività relativa ed una tipologia che classifica tale attività, e la voce di spesa può essere di tipo preventivo (con un valore minimo ed un valore massimo addebitabile al cliente) o di tipo consuntivo (con un monte ore erogato ed un corrispondente valore da addebitare). Inoltre la singola voce di spesa ha un flag che indica se è stata o meno già interamente fatturata.

La singola fattura, associata ad uno ed un sol cliente ed identificata univocamente da un anno di competenza ed un numero progressivo all'interno dell'anno, è composta da una serie di voci di fatturazione che sono raccolte in distinti fogli sulla base della pratica a cui fanno riferimento.

Ogni singola voce di fatturazione di un foglio fa poi riferimento ad una particolare voce di spesa della pratica a cui si riferisce il foglio a cui appartiene, oltre a riportare una descrizione, l'importo da fatturare e l'aliquota IVA.

### Individuare le entità principali

Si vuole realizzare un DB per gestire la fatturazione di una **società**.

L'attività principale dell'azienda è gestire **pratiche** tecnico-legali per i propri **clienti**.

Una singola **pratica** è associata a più **clienti** ed ha un numero di riferimento interno che la identifica univocamente, una data apertura ed uno stato operativo.

Ogni **cliente** può avere più **indirizzi** associati, ma se sono più di uno allora uno ed uno solo tra questi sarà indicato come indirizzo di fatturazione.

Il computo degli **importi da fatturare** è fatto a partire da **voci di spesa** legate alle singole **pratiche**, dove ogni **voce di spesa** ha associato un **operatore** che ha svolto **l'attività** relativa ed una **tipologia** che classifica tale **attività**, e la **voce di spesa** può essere di tipo **preventivo** (con un valore minimo ed un valore massimo addebitabile al cliente) o di tipo **consuntivo** (con un monte ore erogato ed un corrispondente valore da addebitare). Inoltre la singola **voce di spesa** ha un flag che indica se è stata o meno già interamente fatturata.

La singola **fattura**, associata ad uno ed un sol **cliente** ed identificata univocamente da un anno di competenza ed un numero progressivo all'interno dell'anno, è composta da una serie di **voci di fatturazione** che sono raccolte in distinti **fogli** sulla base della **pratica** a cui fanno riferimento.

Ogni singola **voce di fatturazione** di un **foglio** fa poi riferimento ad una particolare **voce di spesa** della pratica a cui si riferisce il **foglio** a cui appartiene, oltre a riportare una descrizione, l'importo da fatturare e l'aliquota IVA.

Abbiamo quindi identificato le seguenti possibili entità:

- società
- pratiche
- clienti
- indirizzi
- importi da fatturare
- voci di spesa
- operatore

- attività
- tipologia
- voce di preventivo
- voce di consuntivo
- fattura
- voce di fatturazione
- fogli

## Verificare se non si sono entità duplicate

Importi da fatturare – Voci di spesa

Voci di spesa - Attività

Decidiamo di utilizzare il termine **Voce di spesa**

## Individuare le associazioni tra le entità

Si vuole realizzare un DB per gestire la fatturazione di una **società**.

L'attività principale dell'azienda è gestire **pratiche** tecnico-legali per i propri clienti.

Una singola **pratica** è associata a più clienti ed ha un numero di riferimento interno che la identifica univocamente, una data apertura ed uno stato operativo.

Ogni **cliente** può avere più **indirizzi** associati, ma se sono più di uno allora uno ed uno solo tra questi sarà indicato come indirizzo di fatturazione.

Il computo degli **importi da fatturare** è fatto a partire da **voci di spesa** legate alle singole **pratiche**, dove ogni **voce di spesa** ha associato un **operatore** che ha svolto **l'attività** relativa ed una **tipologia** che classifica tale **attività**, e la **voce di spesa** può essere di tipo **preventivo** (con un valore minimo ed un valore massimo addebitabile al cliente) o di tipo **consuntivo** (con un monte ore erogato ed un corrispondente valore da addebitare). Inoltre la singola **voce di spesa** ha un flag che indica se è stata o meno già interamente fatturata.

La singola **fattura**, associata ad uno ed un sol cliente ed identificata univocamente da un anno di competenza ed un numero progressivo all'interno dell'anno, è composta da una serie di **voci di fatturazione** che sono raccolte in distinti **fogli** sulla base della **pratica** a cui fanno riferimento.

Ogni singola **voce di fatturazione** di un **foglio** fa poi riferimento ad una particolare **voce di spesa** della pratica a cui si riferisce il **foglio** a cui appartiene, oltre a riportare una descrizione, l'importo da fatturare e l'aliquota IVA.

Possiamo quindi identificare le seguenti associazioni:

- **gestire** pratiche tecnico-legali per i propri clienti
- **pratica** è associata a più clienti
- **cliente** avere più indirizzi
- **cliente** indicato indirizzo di fatturazione
- **voci di spesa** legate alle singole pratiche
- **voce di spesa** ha associato operatore
- **tipologia** classifica attività
  - **voce di spesa** può essere di tipo preventivo
  - **voce di spesa** può essere di tipo consuntivo
- **fattura**, associata ad uno ed un sol cliente
- **fattura**, composta da voci di fatturazione
- **voci di fatturazione** sono raccolte in fogli
- **fogli** sulla base della pratica cui fanno riferimento
- **voce di fatturazione** fa riferimento a voce di spesa

## Verificare se alcune associazioni hanno attributi

In questo caso non ci sono attributi legati ad associazioni.

## Calcolare la cardinalità delle associazioni

- **gestire** pratiche tecnico-legali *per i propri* clienti - *N ad N*
- pratica *è associata a più* clienti - *N ad N*
- cliente *avere* più indirizzi - *1 ad N*
- cliente *indicato* indirizzo di fatturazione - *1 ad 1*
- voci di spesa *legate alle* singole pratiche - *N ad 1*
- voce di spesa *ha associato* operatore - *N ad N*
- tipologia *classifica* attività - *1 ad N*
  - voce di spesa può *essere* di tipo preventivo (*gerarchia*)
  - voce di spesa può *essere* di tipo consuntivo (*gerarchia*)
- fattura, *associata ad uno ed un sol* cliente - *N ad 1*
- fattura, *composta da* voci di fatturazione - *1 ad N*
- voci di fatturazione sono *raccolte* in fogli - *N ad 1*
- pratica a *cui fanno riferimento* le voci di fatturazione - *1 ad N*
- voce di fatturazione *fa riferimento* a voce di spesa - *N ad 1*

## Trovare gli attributi espliciti delle entità

Si vuole realizzare un DB per gestire la fatturazione di una **società**.

L'attività principale dell'azienda è gestire **pratiche** tecnico-legali per i propri **clienti**.

Una singola **pratica** è associata a più **clienti** ed ha un numero di riferimento interno che la identifica univocamente, una data apertura ed uno stato operativo.

Ogni **cliente** può avere più **indirizzi** associati, ma se sono più di uno allora uno ed uno solo tra questi sarà indicato come indirizzo di fatturazione.

Il computo degli **importi da fatturare** è fatto a partire da **voci di spesa** legate alle singole **pratiche**, dove ogni **voce di spesa** ha associato un **operatore** che ha svolto **l'attività** relativa ed una **tipologia** che classifica tale **attività**, e la **voce di spesa** può essere di tipo **preventivo** (con un valore minimo ed un valore massimo addebitabile al cliente) o di tipo **consuntivo** (con un monte ore erogato ed un corrispondente valore da addebitare). Inoltre la singola **voce di spesa** ha un flag che indica se è stata o meno già interamente fatturata.

La singola **fattura**, associata ad uno ed un sol **cliente** ed identificata univocamente da un anno di competenza ed un numero progressivo all'interno dell'anno, è composta da una serie di **voci di fatturazione** che sono raccolte in distinti **fogli** sulla base della **pratica** a cui fanno riferimento.

Ogni singola **voce di fatturazione** di un **foglio** fa poi riferimento ad una particolare **voce di spesa** della pratica a cui si riferisce il **foglio** a cui appartiene, oltre a riportare una descrizione, l'importo da fatturare e l'aliquota IVA.

Abbiamo quindi i seguenti attributi:

Pratica

- numero rif. Interno
- data apertura
- stato operativo

Voce di spesa

- interamente fatturata

Voce di preventivo

- valore minimo
- valore massimo

Voce di consuntivo

- monte ore erogato
- valore da addebitare

Fattura

- anno di competenza
- numero progressivo

Voce di fatturazione

- descrizione
- importo
- aliquota IVA

## **Definire le chiavi primarie delle entità**

Nel testo sono indicate in modo esplicito le seguenti chiavi primarie:  
Abbiamo quindi i seguenti attributi:

Pratica

- numero rif. Interno

Fattura

- anno di competenza
- numero progressivo

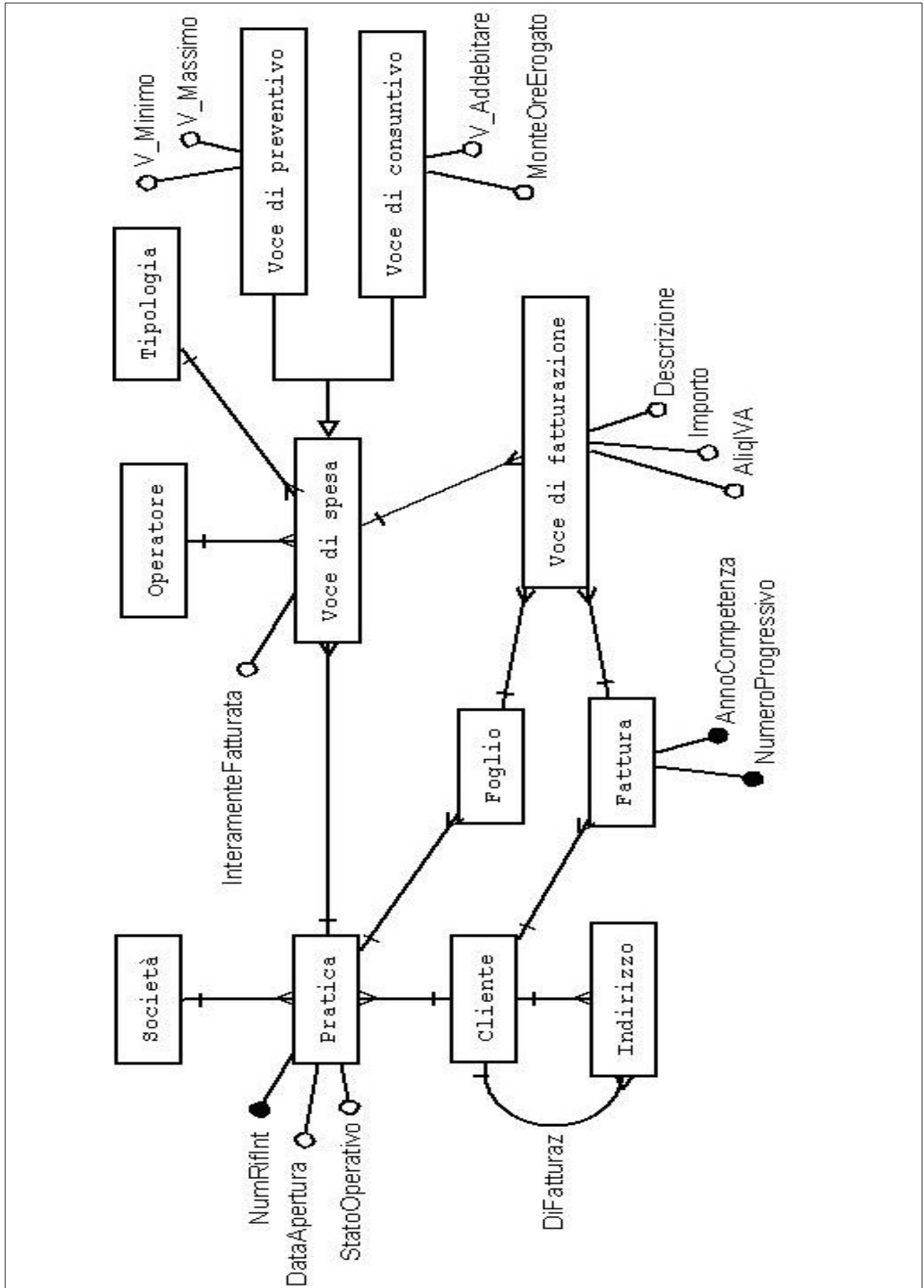
## **Identificazione di eventuali gerarchie**

In questo esempio possono essere identificate le seguenti gerarchie:

Voce di spesa

- Preventiva
- Consuntiva

# Stesura dello schema E-R





## **Esempio 5 : Elezioni amministrative**

### **Testo di partenza**

In occasione delle ultime elezioni amministrative è stato richiesto alla vostra ditta di realizzare il sistema informativo per la gestione dei seggi elettorali.

Per ogni seggio, identificato da un numero univoco all'interno della provincia e da un indirizzo di ubicazione, viene fornita la lista di tutti gli elettori che possono esercitare il loro diritto di voto presso tale seggio, oltre alla lista dei componenti della commissione scrutinante.

E' inoltre necessario registrare l'orario di apertura e di chiusura del seggio per ogni giorno di durata delle operazioni elettorali.

Per ogni elettore si hanno nome, cognome, sesso, data di nascita, luogo di nascita e indirizzo di residenza.

Inoltre per ogni elettore è necessario registrare se ha votato per regione, provincia, comune ed eventualmente zona.

Per ogni membro della commissione scrutinante si devono registrare nome, cognome, luogo di nascita, data di nascita, indirizzo di residenza e ruolo all'interno della commissione (presidente, segretario, scrutatore), nonché se si è presentato o meno a svolgere il servizio.

Nel registro degli elettori è necessario registrare per ogni votante il tipo e numero del documento d'identità, nonché a fini statistici l'ora in cui si è presentato al seggio.

Si hanno poi le liste ed i candidati, dove per ogni tipo di elezione (regionale, provinciale, etc.) ogni lista ha un nome ed un simbolo, mentre ogni candidato appartiene ad una sola lista ed è identificato da nome, cognome, data e luogo di nascita.

Inoltre alcune liste sono tra loro raggruppate, ed è quindi necessario indicare, al fine di poter produrre gli opportuni report, se ed a quale altra lista di riferimento è collegata una lista.

A fine votazione avviene lo scrutinio, a seguito del quale è necessario registrare quanti voti sono stati assegnati ad ogni candidato e quanti ad ogni lista (una lista può raccogliere voti che indicano solo la lista ma non un candidato), nonché quanti sono stati i voti nulli, le schede bianche e quelle contestate.

### **Individuare le entità**

In occasione delle ultime elezioni amministrative è stato richiesto alla vostra ditta di realizzare il sistema informativo per la gestione dei **seggi elettorali**.

Per ogni **seggio**, identificato da un numero univoco all'interno della provincia e da un indirizzo di ubicazione, viene fornita la **lista di tutti gli elettori** che possono esercitare il loro diritto di voto presso tale **seggio**, oltre alla **lista dei componenti della commissione scrutinante**.

E' inoltre necessario registrare l'orario di apertura e di chiusura del **seggio** per ogni **giorno di durata delle operazioni elettorali**.

Per ogni **elettore** si hanno nome, cognome, sesso, data di nascita, luogo di nascita e indirizzo di residenza.

Inoltre per ogni **elettore** è necessario registrare se ha votato per regione, provincia, comune ed eventualmente zona.

Per ogni **membro della commissione scrutinante** si devono registrare nome, cognome, luogo di nascita, data di nascita, indirizzo di residenza e ruolo all'interno della commissione (presidente, segretario, scrutatore), nonché se si è presentato o meno a svolgere il servizio.

Nel **registro degli elettori** è necessario registrare per ogni **votante** il tipo e numero del documento d'identità, nonché a fini statistici l'ora in cui si è presentato al **seggio**.

Si hanno poi le **liste** ed i **candidati**, dove per ogni **tipo di elezione** (regionale, provinciale, etc.) ogni **lista** ha un nome ed un simbolo, mentre ogni **candidato** appartiene ad una sola **lista** ed è identificato da nome, cognome, data e luogo di nascita.

Inoltre alcune **liste** sono tra loro raggruppate, ed è quindi necessario indicare, al fine di poter produrre gli opportuni report, se ed a quale altra **lista di riferimento** è collegata una **lista**.

A fine votazione avviene lo scrutinio, a seguito del quale è necessario registrare quanti **voti** sono stati assegnati ad ogni **candidato** e quanti ad ogni **lista** (una lista può raccogliere voti che indicano solo la lista ma non un candidato), nonché quanti sono stati i **voti** nulli, le schede bianche e quelle contestate.

Abbiamo quindi identificato le seguenti possibili entità:

- **seggi elettorali.**
- **lista di tutti gli elettori**
- **elettori**
- **lista dei componenti della commissione scrutinante**
- **commissione scrutinante**
- **giorno di durata delle operazioni elettorali**
- **membro della commissione scrutinante**
- **registro degli elettori**
- **votante**
- **elezione**
- **tipo di elezione**
- **liste**
- **candidati**
- **lista di riferimento**
- **voti**

## **Verificare se non si sono entità duplicate**

Abbiamo quindi identificato le seguenti possibili entità duplicate:

lista di tutti gli elettori - registro degli elettori  
Decidiamo di usare il termine “Registro”

Elettori - votanti  
Decidiamo di usare il termine “Elettore”

Lista componenti della commissione scrutinante - commissione scrutinante  
Decidiamo di usare il termine “Commissione”

componenti della commissione scrutinante - membro della commissione scrutinante  
Decidiamo di usare il termine “Membro”

elezione – tipo di elezione  
Decidiamo di usare il termine “Elezione”

## **Eventuale cambio dei nomi per semplicità**

Semplifichiamo i seguenti nomi:

Seggi elettorali	-> Seggio
Giorno di durata delle operazioni elettorali	-> Giorno

Lista finale delle entità:

- **seggio**
- **registro**
- **elettore**
- **commissione**
- **giorno**
- **membro**
- **elezione**
- **lista**
- **candidato**
- **voto**

## Individuare le associazioni tra le entità

In occasione delle ultime elezioni amministrative è stato richiesto alla vostra ditta di realizzare il sistema informativo per la gestione dei **seggi elettorali**.

Per ogni **seggio**, identificato da un numero univoco all'interno della provincia e da un indirizzo di ubicazione, viene fornita la **lista di tutti gli elettori** che possono esercitare il loro diritto di voto presso tale **seggio**, oltre alla **lista dei componenti della commissione scrutinante**.

E' inoltre necessario registrare l'orario di apertura e di chiusura del **seggio per ogni giorno di durata delle operazioni elettorali**.

Per ogni **elettore** si hanno nome, cognome, sesso, data di nascita, luogo di nascita e indirizzo di residenza.

Inoltre per ogni **elettore** è necessario registrare se ha votato per regione, provincia, comune ed eventualmente zona.

Per ogni **membro della commissione scrutinante** si devono registrare nome, cognome, luogo di nascita, data di nascita, indirizzo di residenza e ruolo all'interno della commissione (presidente, segretario, scrutatore), nonché se si è presentato o meno a svolgere il servizio.

Nel **registro degli elettori** è necessario registrare per ogni **volante** il tipo e numero del documento d'identità, nonché a fini statistici l'ora in cui si è presentato al **seggio**.

Si hanno poi le **liste** ed i **candidati**, dove per ogni **tipo di elezione** (regionale, provinciale, etc.) ogni **lista** ha un nome ed un simbolo, mentre ogni **candidato** appartiene ad una sola lista ed è identificato da nome, cognome, data e luogo di nascita.

Inoltre alcune **liste** sono tra loro raggruppate, ed è quindi necessario indicare, al fine di poter produrre gli opportuni report, se ed a quale altra **lista di riferimento è collegata una lista**.

A fine votazione avviene lo scrutinio, a seguito del quale è necessario registrare quanti **voti sono stati assegnati ad ogni candidato** e quanti ad ogni lista (una lista può raccogliere voti che indicano solo la lista ma non un candidato), nonché quanti sono stati i **voti nulli**, le schede bianche e quelle contestate.

Possiamo quindi identificare le seguenti associazioni:

- **registro di tutti gli elettori**
- **seggio** viene fornita registro
- **seggio**, viene fornita la commissione
- **membro della commissione**
- **seggio per ogni giorno**
- **liste** ha un nome ed un simbolo per ogni elezione
- **candidato** appartiene ad una sola lista
- **lista di riferimento** è collegata una lista.
- **voti** sono stati assegnati ad ogni candidato
- **voti** sono stati assegnati ad ogni lista

Vi è poi una relazione implicita nella frase “per ogni **elettore** è necessario registrare se ha votato per regione, provincia, comune ed eventualmente zona”, che implica una relazione tra elettore e tipo di elezione: **elettore** ha votato per elezione

## Verificare se alcune associazioni hanno attributi

*liste* ha un nome ed un simbolo per ogni tipo di elezione

Quindi per la relazione tra lista ed elezione vi sono gli attributi NOME e SIMBOLO.

Questi attributi possono essere spostati nell'entità lista, creando liste distinte per ogni tipologia di elezione, in considerazione anche del fatto (non espresso dal testo ma noto ad ogni elettore di buon senso) che un candidato dovrebbe potersi presentare solo per un certo tipo di elezione (se si presenta per la provincia non dovrebbe presentarsi per il comune).

## Calcolare la cardinalità delle associazioni

- **seggio** viene fornita la registro - 1 ad 1
- **seggio**, viene fornita la commissione - 1 ad 1
- **seggio** per ogni giorno - 1 ad N
- **liste** ha un nome ed un simbolo per ogni tipo di elezione - N ad 1
- **candidato** appartiene ad una sola lista - N ad 1
- **lista di riferimento** è collegata una lista - 1 ad N
- **voti** sono stati assegnati ad ogni candidato - N ad 1
- **voti** sono stati assegnati ad ogni lista - N ad 1
- **elettore** ha votato per elezione - N ad N

La relazione tra elettore e tipo di elezione è molti a molti, quindi verrà risolta come una entità (haVotato) di correlazione tra le due entità di base.

Le tre entità seggio, registro e commissione sono in relazione 1 ad 1 tra loro, e quindi possono essere collassate in un'unica entità.

## Trovare gli attributi espliciti delle entità

In occasione delle ultime elezioni amministrative è stato richiesto alla vostra ditta di realizzare il sistema informativo per la gestione dei **seggi elettorali**.

Per ogni **seggio**, identificato da un numero univoco all'interno della provincia e da un indirizzo di ubicazione, viene fornita la **lista di tutti gli elettori** che possono esercitare il loro diritto di voto presso tale **seggio**, oltre alla **lista dei componenti della commissione scrutinante**.

E' inoltre necessario registrare l'orario di apertura e di chiusura del **seggio** per ogni **giorno di durata delle operazioni elettorali**.

Per ogni **elettore** si hanno nome, cognome, sex, data di nascita, luogo di nascita e indirizzo di residenza.

Inoltre ogni **elettore** è necessario registrare se ha votato per regione, provincia, comune ed eventualmente zona.

Per ogni **membro della commissione scrutinante** si devono registrare nome, cognome, luogo di nascita, data di nascita, indirizzo di residenza e ruolo all'interno della commissione (presidente, segretario, scrutatore), nonché se si è presentato o meno a svolgere il servizio.

Nel **registro degli elettori** è necessario registrare per ogni **votante** il tipo e numero del documento d'identità, nonché a fini statistici l'ora in cui si è presentato al **seggio**.

Si hanno poi le **liste** ed i **candidati**, dove per ogni tipo di **elezione** (regionale, provinciale, etc.) ogni **lista** ha un nome ed un simbolo, mentre ogni **candidato** appartiene ad una sola **lista** ed è identificato da nome, cognome, data e luogo di nascita.

Inoltre alcune **liste** sono tra loro raggruppate, ed è quindi necessario indicare, al fine di poter produrre gli opportuni report, se ed a quale altra **lista di riferimento** è collegata una **lista**.

A fine votazione avviene lo scrutinio, a seguito del quale è necessario registrare quanti **voti** sono stati assegnati ad ogni **candidato** e quanti ad ogni **lista** (una lista può raccogliere voti che indicano solo la lista ma non un candidato), nonché quanti sono stati i **voti nulli**, le schede bianche e quelle contestate.

## Definire le chiavi primarie delle entità

Nel testo sono indicate in modo esplicito le seguenti chiavi primarie :

Seggio:

- numero univoco
- provincia

Candidato:

- nome
- cognome
- data di nascita
- luogo di nascita

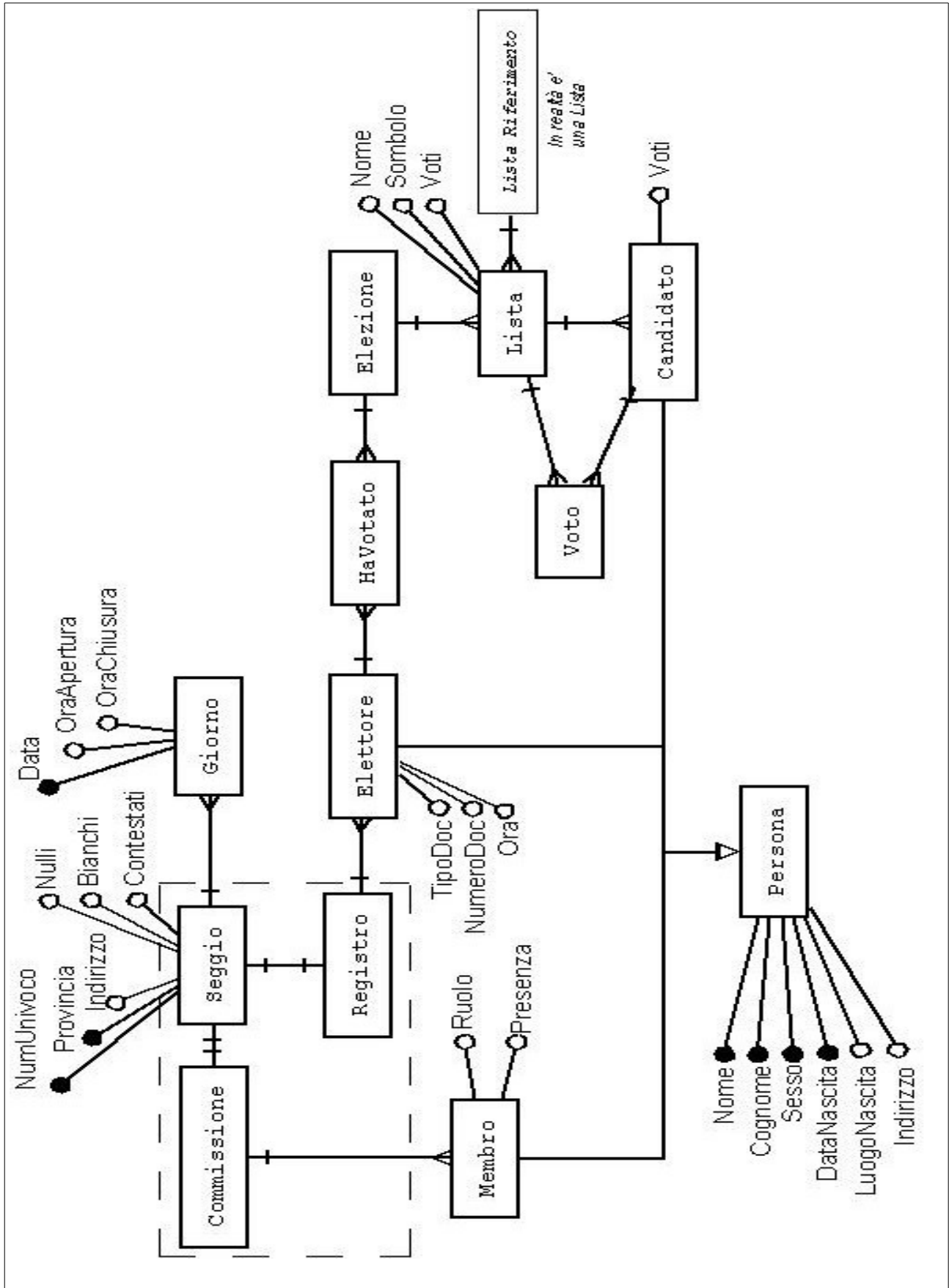
## Identificazione di eventuali gerarchie

In questo esempio possono essere identificate le seguenti gerarchie:

Persona

- Elettore
- Candidato
- Membro

# Stesura dello schema E-R



## **Esempio X : xxx Scuole elementari**

### **Testo di partenza**

### **Individuare le entità**

Abbiamo quindi identificato le seguenti possibili entità:

- scuole

### **Verificare se non si sono entità duplicate**

### **Individuare le associazioni tra le entità**

Possiamo quindi identificare le seguenti associazioni:

- *scuola* è composta da *classi*

### **Verificare se alcune associazioni hanno attributi**

In questo caso non ci sono attributi legati ad associazioni.

### **Calcolare la cardinalità delle associazioni**

- *scuola* è composta da *classi* - 1 ad N
- 

### **Trovare gli attributi espliciti delle entità**

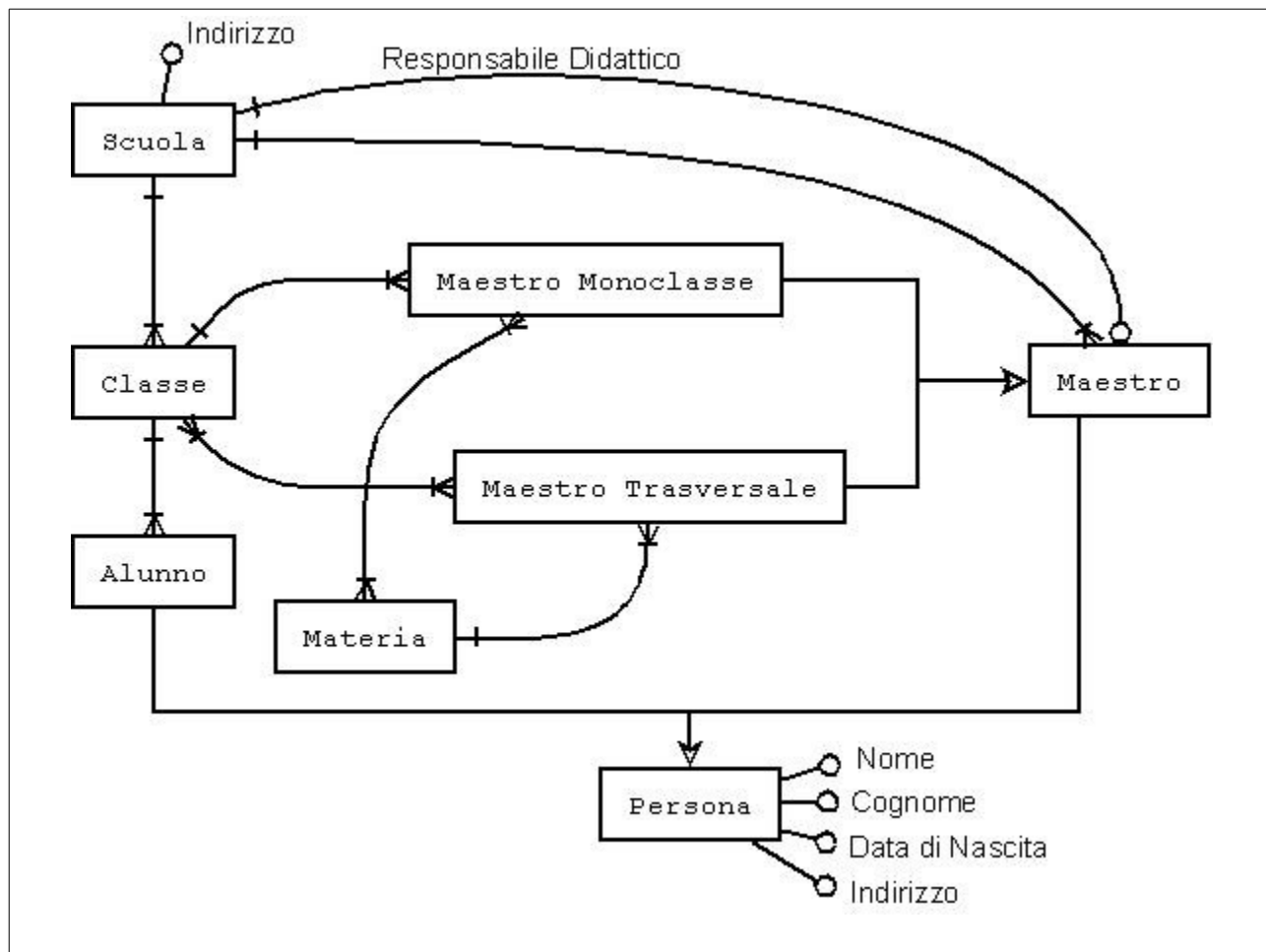
### **Definire le chiavi primarie delle entità**

Nel testo non sono indicate in modo esplicito le chiavi primarie delle entità.

### **Identificazione di eventuali gerarchie**

In questo esempio possono essere identificate le seguenti gerarchie:

## Stesura dello schema E-R





# **PATTERN DI MODELLAZIONE**

*Introduzione ai Pattern*

*Master-Detail*