



# Capitolo 7

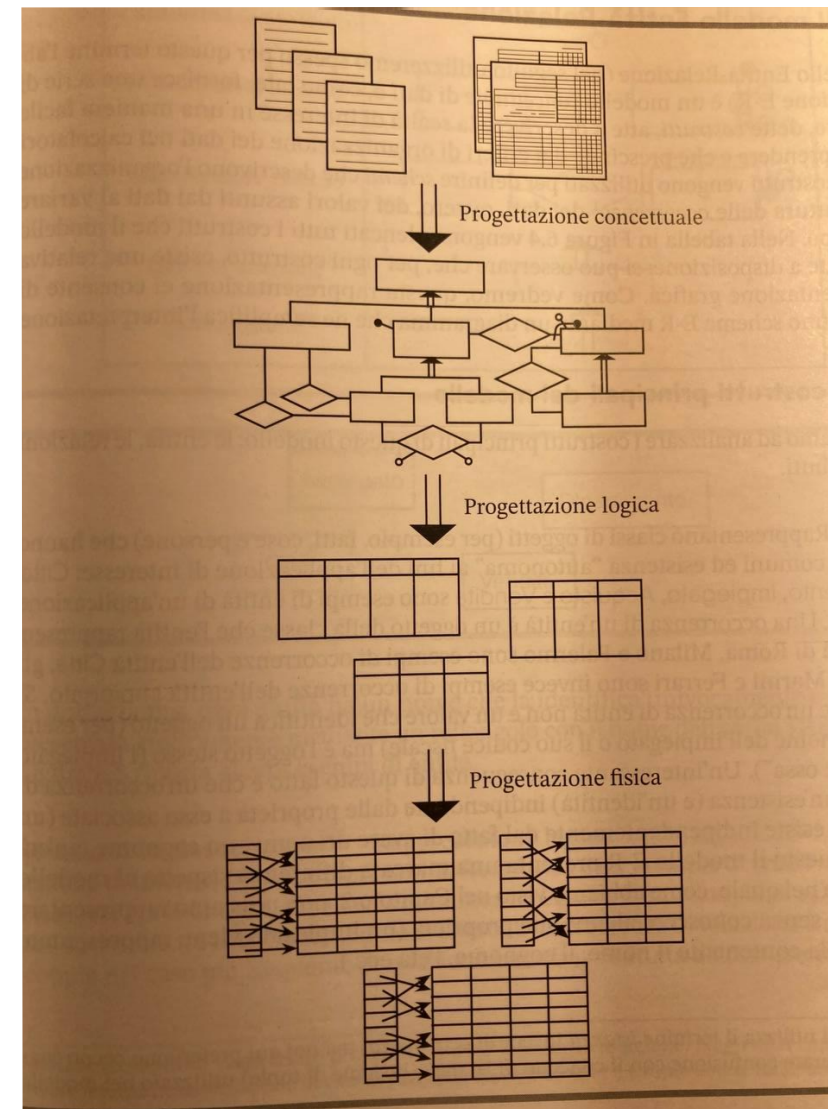
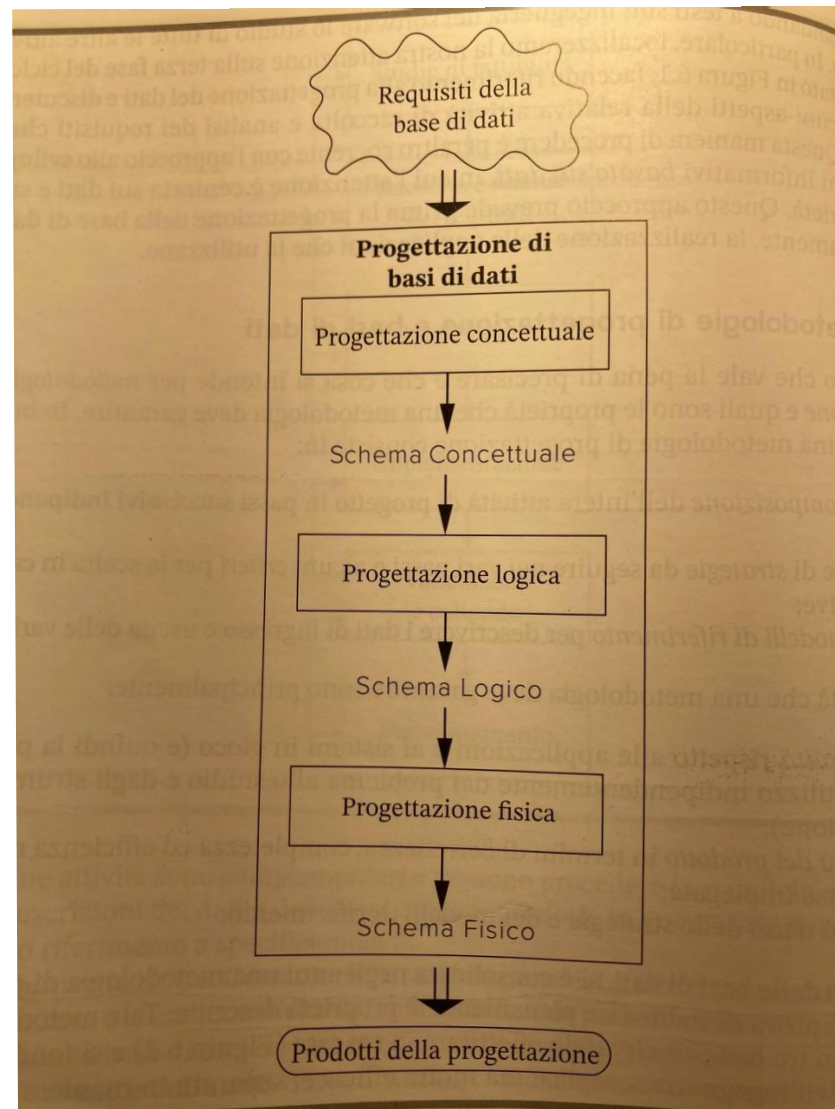
## Progettazione concettuale

# Cos'è la progettazione concettuale

- Risponde al problema di *progettare* una base di dati a partire dai suoi *requisiti*
- Progettare una base di dati significa definirne
  - Struttura
  - Caratteristiche
  - Contenuto
- Presenteremo una metodologia di progettazione che si è *largamente* diffusa nell'ambito della progettazione delle basi di dati
  - Tale metodologia è articolata in tre fasi



## Metodologia di progettazione di una base di dati



# Cos'è la progettazione concettuale

- Il suo scopo è quello di rappresentare le specifiche informali della realtà di interesse
- È *indipendente* dai criteri di rappresentazione usati dai DBMS
- Il prodotto di questa fase viene chiamato *modello concettuale*
- In questa fase, il progettista deve rappresentare il *contenuto informativo* della base di dati, senza preoccuparsi né delle modalità di rappresentazione né dell'efficienza



# Cos'è la progettazione logica

- Consiste nella traduzione dello schema concettuale in un modello di rappresentazione dei dati adottato dal sistema di gestione della base di dati a disposizione
- Il prodotto di questa fase viene chiamato *schema logico*
- E' ancora indipendente dai dettagli fisici
- Possono essere comunque presenti alcuni criteri di ottimizzazione delle operazioni da effettuare, e controlli formali di qualità:
  - Nel caso del modello relazionale, la tecnica comunemente usata è quella della normalizzazione



# Cos'è la progettazione fisica

- In questa fase, lo schema logico viene completato con la specifica dei parametri fisici di memorizzazione dei dati
  - I dati vengono descritti da un punto di vista fisico (tipo e dimensione dei campi)
  - Definizione degli indici
- Il prodotto di questa fase viene detto *schema fisico*
- Tale modello dipende dello specifico DBMS





# Analisi dei requisiti e progettazione concettuale ("Analisi dei dati")

- Comprende attività (interconnesse) di
  - acquisizione dei requisiti
  - analisi dei requisiti
  - costruzione del glossario
  - costruzione dello schema concettuale



## Requisiti

- Possibili fonti:
  - **Utenti e committenti**, attraverso:
    - interviste
    - documentazione apposita
  - **documentazione esistente**:
    - normative (leggi, regolamenti di settore)
    - regolamenti interni, procedure aziendali
    - realizzazioni preesistenti
  - **modulistica**





# Acquisizione e analisi dei requisiti

- Il reperimento dei requisiti è un'attività difficile e non standardizzabile
- l'attività di analisi inizia con i primi requisiti raccolti e spesso indirizza verso altre acquisizioni



# Acquisizione per interviste



- utenti diversi possono fornire informazioni diverse
- utenti a livello più alto hanno spesso una visione più ampia ma meno dettagliata
- le interviste portano spesso ad una acquisizione dei requisiti “per raffinamenti successivi”

# Interazione con gli utenti

- La specifica dei requisiti raccolti avviene spesso facendo uso di descrizioni in *linguaggio naturale*.
  - Fonte di *ambiguità* e fraintendimenti
- Occorre
  - effettuare spesso verifiche di comprensione e coerenza
  - verificare anche per mezzo di esempi (generalisti e relativi a casi limite)
  - richiedere definizioni e classificazioni
  - far evidenziare gli aspetti essenziali rispetto a quelli marginali



# Esempio di requisiti espressi in linguaggio naturale

Si vuole realizzare una base di dati per una società che eroga corsi, di cui vogliamo rappresentare i dati dei partecipanti ai corsi e dei docenti. Per i partecipanti (circa 5000), identificati da un codice, si vuole memorizzare il codice fiscale, il cognome, l'età, il sesso, il luogo di nascita, il nome dei loro attuali datori di lavoro, i posti dove hanno lavorato in precedenza insieme al periodo, l'indirizzo e il numero di telefono, i corsi che hanno frequentato (i corsi sono in tutto circa 200) e il giudizio finale. Rappresentiamo anche i seminari che stanno attualmente frequentando e, per ogni giorno, i luoghi e le ore dove sono tenute le lezioni.

I corsi hanno un codice, un titolo e possono avere varie edizioni con date di inizio e fine e numero di partecipanti. Se gli studenti sono liberi professionisti, vogliamo conoscere l'area di interesse e, se lo possiedono, il titolo. Per quelli che lavorano alle dipendenze di altri, vogliamo conoscere invece il loro livello e la posizione ricoperta. Per gli insegnanti (circa 300), rappresentiamo il cognome, l'età, il posto dove sono nati, il nome del corso che insegnano, quelli che hanno insegnato nel passato e quelli che possono insegnare.

Rappresentiamo anche tutti i loro recapiti telefonici. I docenti possono essere dipendenti interni della società o collaboratori esterni.



# Esempio di requisiti espressi in linguaggio naturale

E' facile rendersi conto che il testo dei requisiti presenta un certo numero di *ambiguità* e impressioni.

- e.g., si utilizzano i termini “*partecipante*” e “*studente*” per indicare lo stesso concetto
- La stessa cosa accade per i termini “*docente*”/”*professore*” e “*corso*”/”*seminario*”



# Requisiti: documentazione descrittiva

- Regole generali per ottenere una specifica dei requisiti più precisa e senza ambiguità:
  - scegliere il corretto livello di astrazione
  - standardizzare la struttura delle frasi
  - suddividere le frasi articolate
  - separare le frasi sui dati da quelle sulle operazioni



# Requisiti: organizzazione di termini e concetti

- Regole generali
  - costruire un glossario dei termini
  - individuare omonimi e sinonimi e unificare i termini
  - rendere esplicito il riferimento fra termini
  - riorganizzare le frasi per concetti





## Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Partecipante	Persona che partecipa ai corsi	Studente	Corso, Società
Docente	Docente dei corsi. Può essere esterno	Insegnante	Corso
Corso	Corso organizzato dalla società. Può avere più edizioni.	Seminario	Docente
Società	Ente presso cui i partecipanti lavorano o hanno lavorato	Posti	Partecipante

# Principali modifiche al nostro esempio

- *Luogo* (di nascita) è omonimo di *Luogo* (dove si tengono le lezioni)
  - Va sostituito con *Città*
- *Posto* (dove hanno lavorato in precedenza)
  - Va sostituito con *Datore di lavoro*
- Il *giudizio* va espresso come *votazione in decimi*
- *Periodo* va interpretato come *data di inizio e fine rapporto*
- *Seminario* va sostituito con *edizione del corso*
- *Luogo* (dove si tengono le lezioni) va sostituito con *aula*
- Ecc...





# Strutturazione dei requisiti in gruppi di frasi omogenee

<b>Fraasi di carattere generale</b>
<i>Si vuole realizzare una base di dati per una società che eroga corsi, di cui vogliamo rappresentare i dati dei partecipanti ai corsi e dei docenti.</i>
<b>Fraasi relative ai partecipanti</b>
<i>Per i partecipanti (circa 5000), identificati da un codice, rappresentiamo il codice fiscale, il cognome, l'età, il sesso, la città di nascita, i nomi dei loro attuali datori di lavoro e di quelli precedenti (insieme alle date di inizio e fine rapporto), le edizioni dei corsi che stanno attualmente frequentando e quelli che hanno frequentato in passato, con la relativa votazione finale in decimi.</i>
<b>Fraasi relative ai datori di lavoro</b>
<i>Relativamente ai datori di lavoro presenti e passati dei partecipanti, rappresentiamo il nome, l'indirizzo e il numero di telefono.</i>
<b>Fraasi relative ai corsi</b>
<i>Per i corsi (circa 180), rappresentiamo il titolo e il codice, le varie edizioni con date di inizio e fine e, per ogni edizione, rappresentiamo il numero di partecipanti e il giorno della settimana, le aule e le ore dove si sono tenute le lezioni.</i>
<b>Fraasi relative a tipi specifici di partecipanti</b>
<i>Per i partecipanti che sono liberi professionisti, rappresentiamo l'area di interesse e, se lo possiedono, il titolo professionale. Per i partecipanti che sono dipendenti, rappresentiamo invece il loro livello e la posizione ricoperta.</i>
<b>Fraasi relative ai docenti</b>
<i>Per i docenti (circa 300), rappresentiamo il cognome, l'età, la città di nascita, tutti i numeri di telefono, il titolo del corso che insegnano, di quelli che hanno insegnato in passato e di quelli che possono insegnare. I docenti possono essere dipendenti interni della società di formazione o collaboratori esterni.</i>





# Operazioni

- **Operazione 1:** inserisci un nuovo partecipante indicando tutti i suoi dati
  - operazione da effettuare in media 40 volte al giorno
- **Operazione 2:** assegna un partecipante a un'edizione di corso
  - Circa 50 volte al giorno
- **Operazione 3:** inserisci un nuovo docente indicando tutti i suoi dati e i corsi che può insegnare
  - 2 volte al giorno
- **Operazione 4:** assegna un nuovo docente abilitato a un'edizione di un corso
  - 15 volte al giorno





# Operazioni

- **Operazione 5:** stampa tutte le informazioni sulle edizioni passate di un corso con titolo, orari lezioni e numero partecipanti
  - 10 volte al giorno
- **Operazione 6:** stampa tutti i corsi offerti, con informazioni sui docenti che possono insegnarli
  - 20 volte al giorno
- **Operazione 7:** per ogni docente, trova i partecipanti a tutti i corsi da lui/lei insegnati
  - 5 volte alla settimana
- **Operazione 8:** effettua una statistica su tutti i partecipanti a un corso con tutte le informazioni su di essi
  - 10 volte al mese



Dopo questa strutturazione dei requisiti, siamo pronti ad avviare la prima fase della progettazione che consiste nella costruzione di uno schema concettuale in grado di descrivere in maniera adeguata *tutte* le specifiche dei dati raccolte.

*Ogni fase rappresenta lo stesso insieme di requisiti, in forma diversa.*

# **Fine dell'analisi dei requisiti**

# **Inizio della rappresentazione**

# **concettuale dei dati**

# **(schema ER)**

# Requisiti discorsivi e schemi concettuali

Prima di affrontare le metodologie di progetto, cerchiamo di stabilire alcune buone pratiche per una corretta rappresentazione concettuale dei dati.

Iniziamo da alcuni criteri generali di rappresentazione, per poi passare a una rassegna di alcuni classici *design pattern*.



Quale costrutto E-R va utilizzato per rappresentare un concetto presente nelle specifiche?

Bisogna basarsi sulle definizioni dei costrutti del modello E-R

- se ha proprietà significative e descrive oggetti con esistenza autonoma
  - **entità**
- se è semplice e non ha proprietà
  - **attributo**
- se correla due o più concetti
  - **relationship**
- se è caso particolare di un altro
  - **generalizzazione**

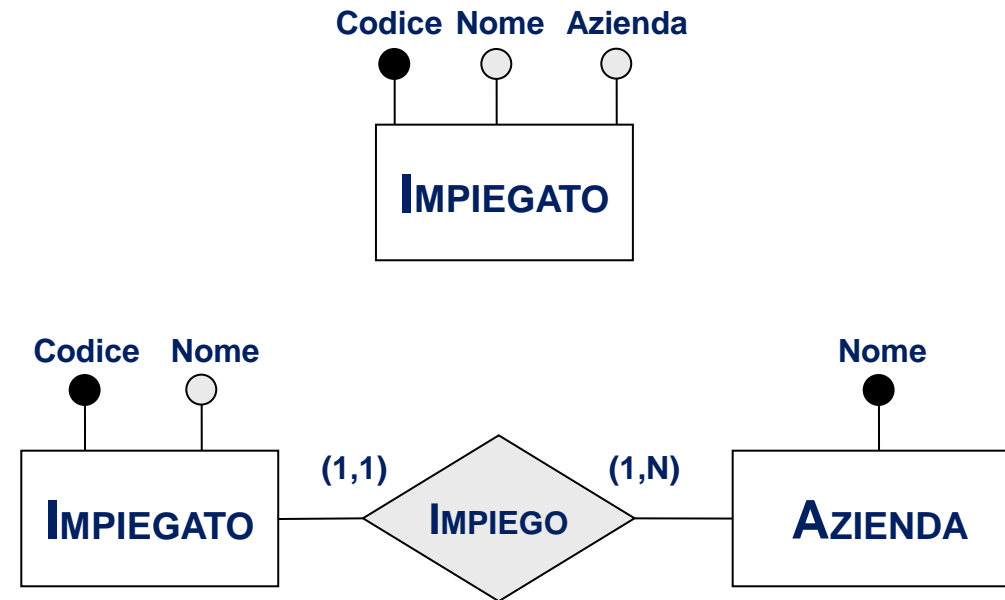
# Design pattern

- Soluzioni progettuali a problemi comuni
- Largamente usati nell'ingegneria del software
- Vediamo alcuni pattern comuni nella progettazione concettuale di basi di dati

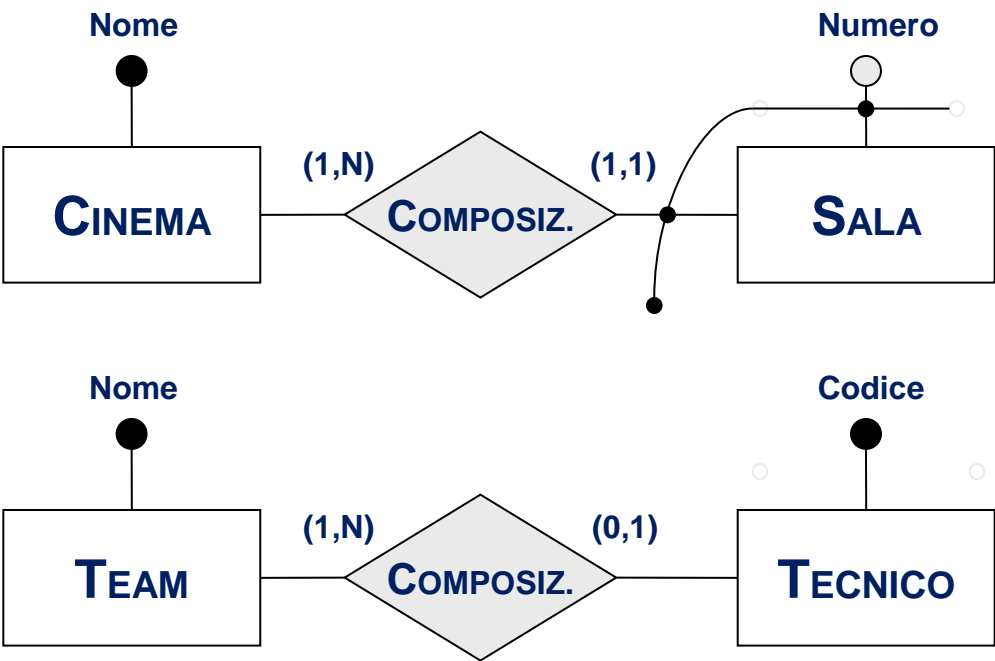




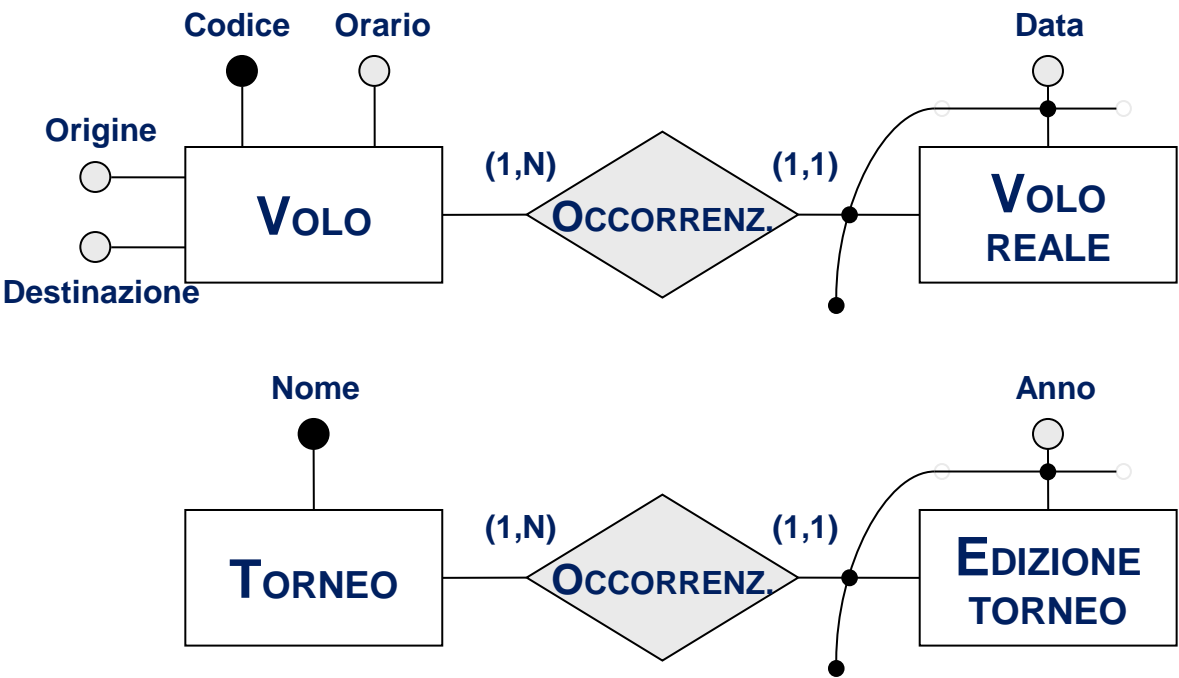
## Reificazione di attributo di entità



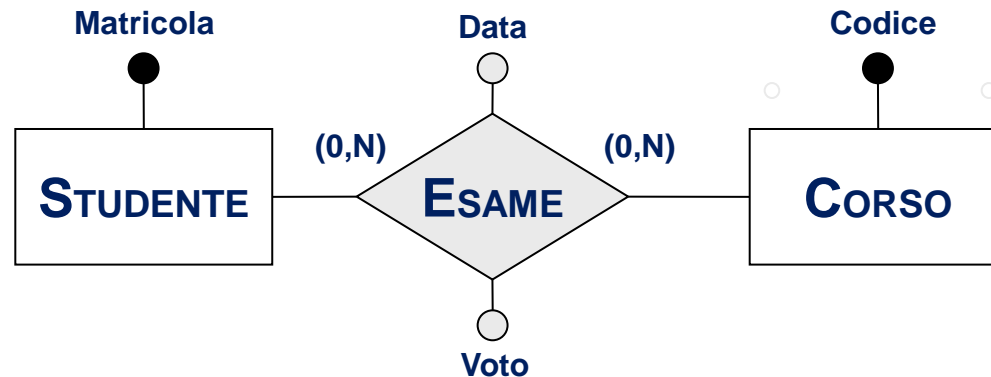
Part-of



# Instance-of



## Reificazione di relazione binaria

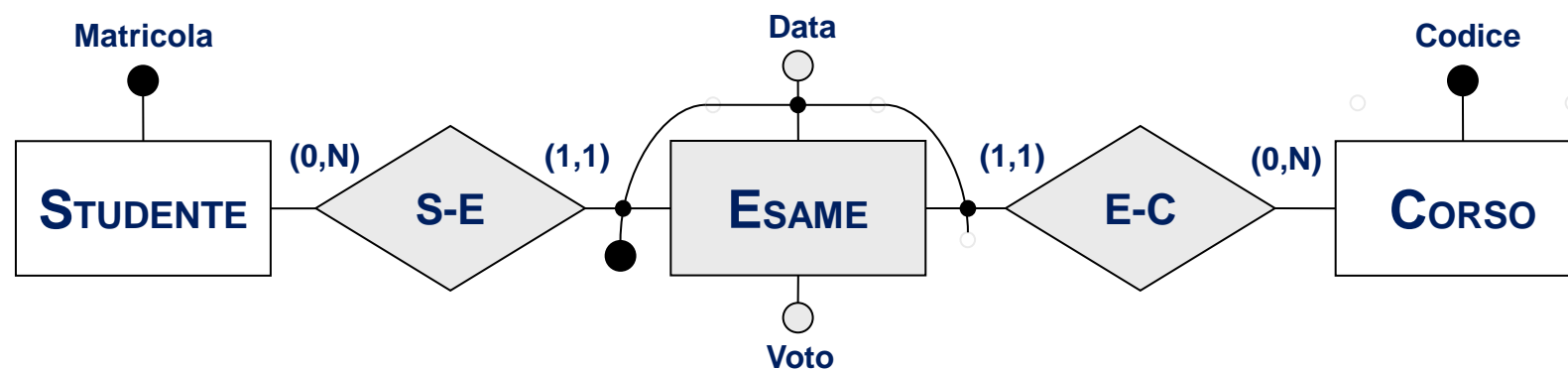


Soluzione valida solo se ogni studente può sostenere una sola volta un certo esame. Perché:

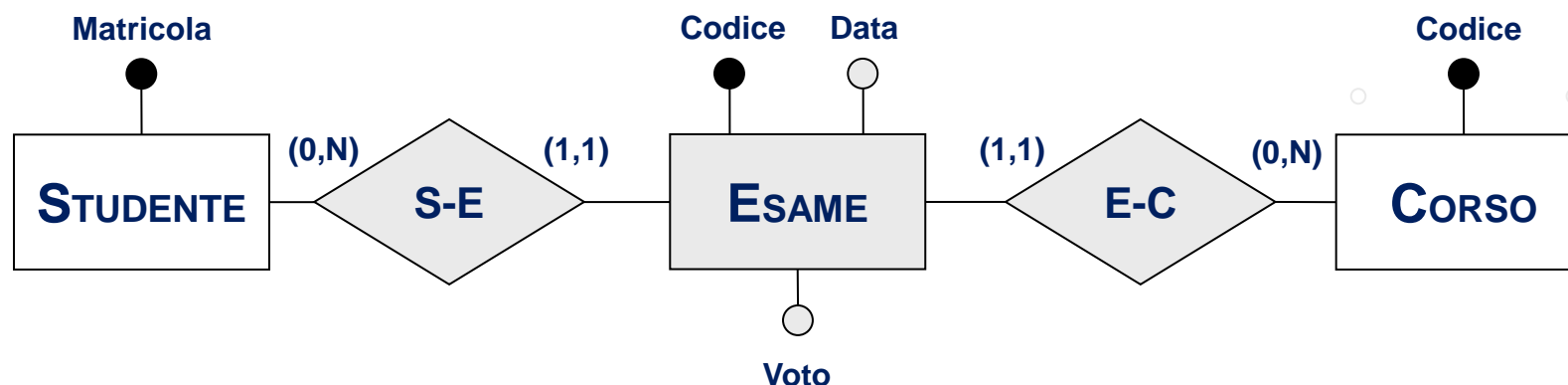
Occorrenze della relazione Esame = insieme di coppie (studente, corso) senza duplicati.

Il fatto che la relazione Esame abbia degli attributi associati non cambia la situazione.

# Reificazione di relazione binaria



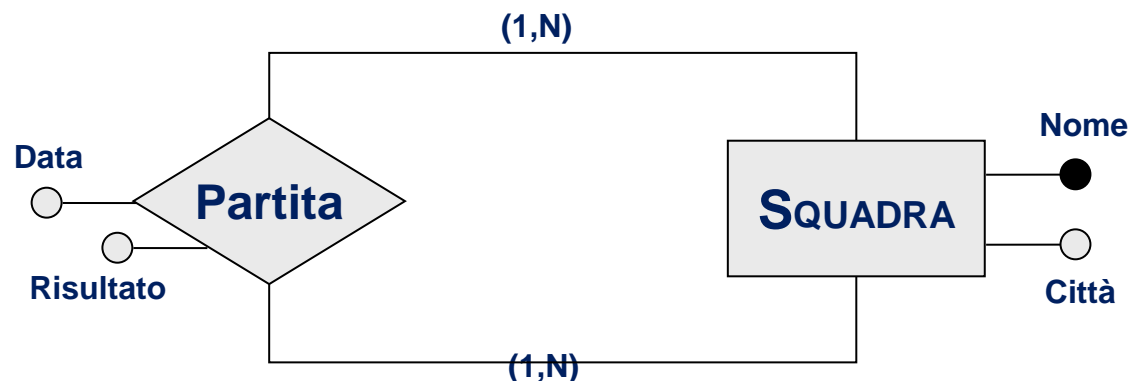
Soluzione corretta:  
L'identificazione di un esame avviene tramite:  
(i) lo studente  
(ii) il corso  
(iii) la data.



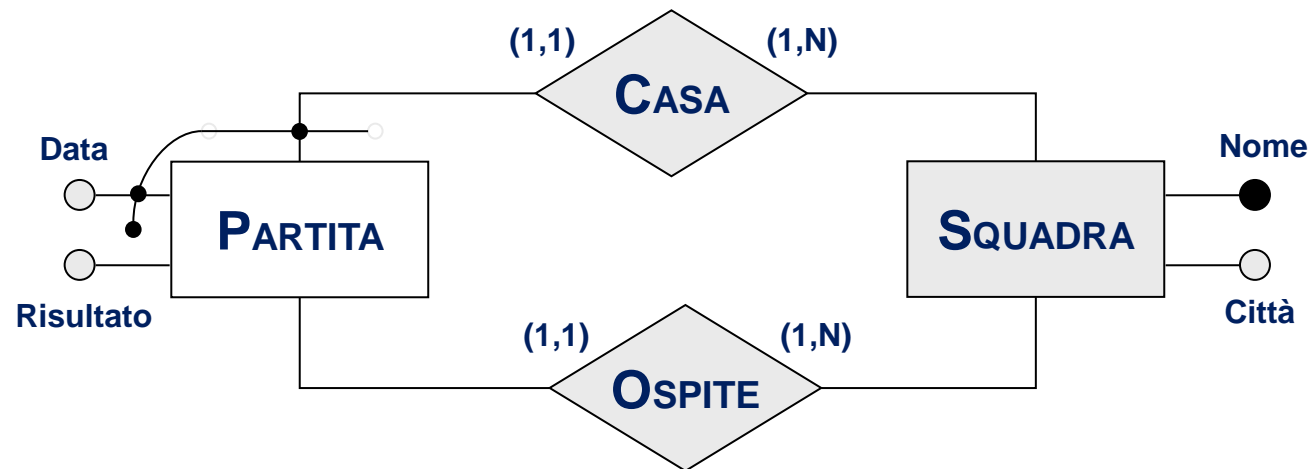
Soluzione alternativa:  
Introduzione di un codice identificativo.  
Svantaggio: il codice è un concetto nuovo non presente nelle specifiche

## Reificazione di relazione ricorsiva

Svantaggio: due squadre  
 non si possono incontrare più  
 di due volte. Possiamo  
 avere al massimo  
 X vs Y  
 Y vs X



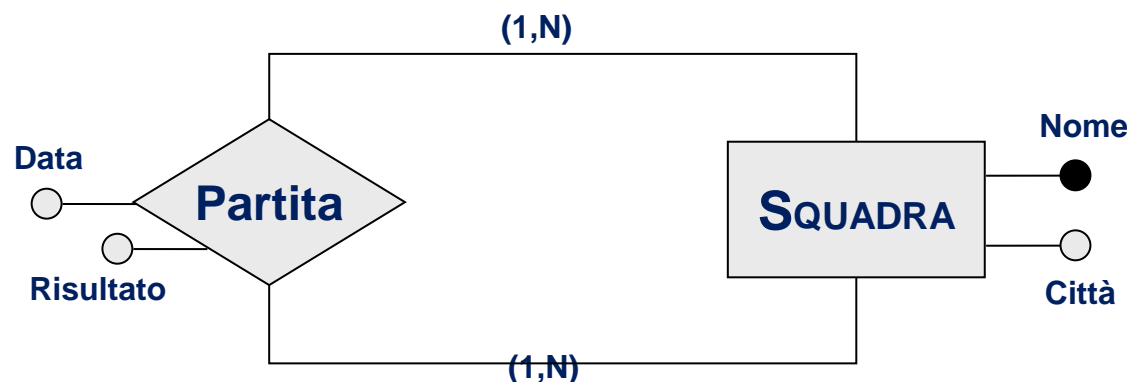
In questa soluzione, abbiamo  
 reificato la relazione Partita, che ora  
 è un'entità identificata dalla Data e  
 dalla squadra che gioca in casa.  
 Ora, *una squadra può giocare solo  
 una partita in una certa data.*



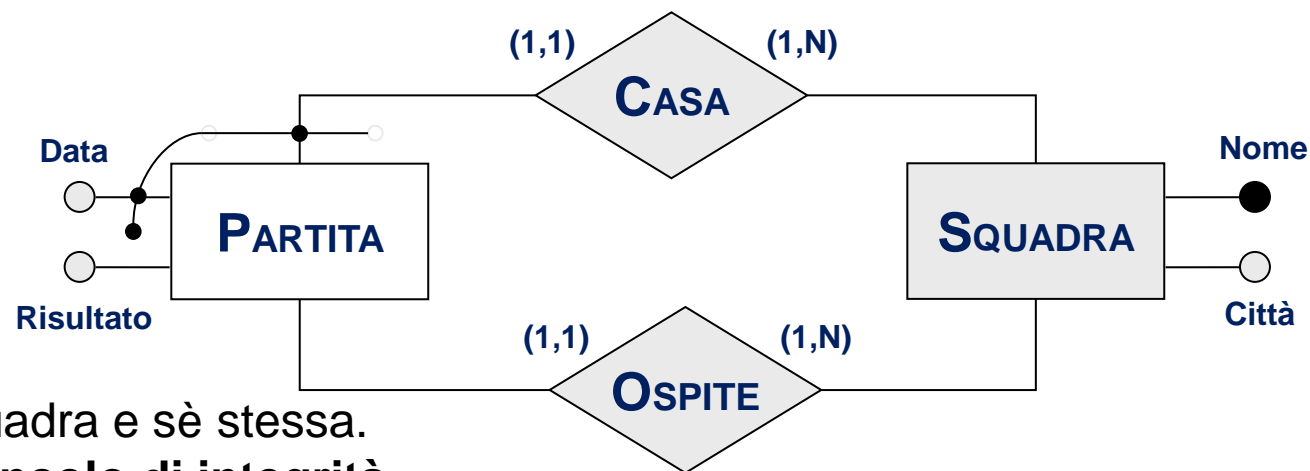


## Reificazione di relazione ricorsiva

Svantaggio: due squadre  
non si possono incontrare più  
di due volte. Possiamo  
avere al massimo  
X vs Y  
Y vs X



In questa soluzione, la squadra X  
può giocare più volte contro Y,  
purché in date diverse. E.g.:  
(23/10/2024, 1-0, X, Y)  
(24/11/2024, 0-2, X, Y)  
(05/12/2024, 1-1, X, Y)



**Attenzione:** ci può essere una partita tra una squadra e se stessa.  
Vincolo non esprimibile in E-R. Aggiungere un **vincolo di integrità**  
scritto in linguaggio naturale sotto lo schema E-R.

## Caso particolare

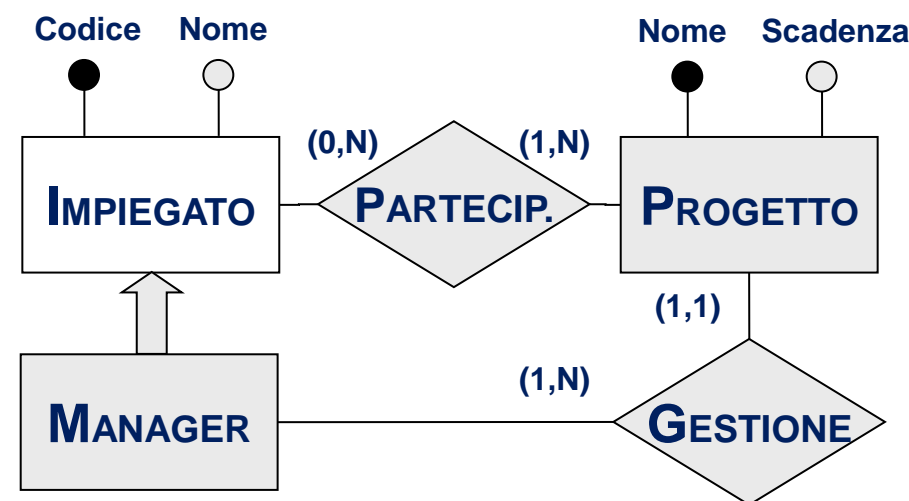
Si vuole rappresentare il sottoinsieme degli impiegati che sono anche manager.

---> uso le generalizzazioni

In questo modo, è possibile specializzare anche i vari ruoli di un impiegato (vedi relazione Gestione).

Attenzione ai cicli: abbiamo anche il vincolo che “ogni manager gestisce un (solo) progetto *al quale partecipa*”.

Quindi ogni coppia (manage, progetto) in Gestione, deve comparire anche in Partecipazione.

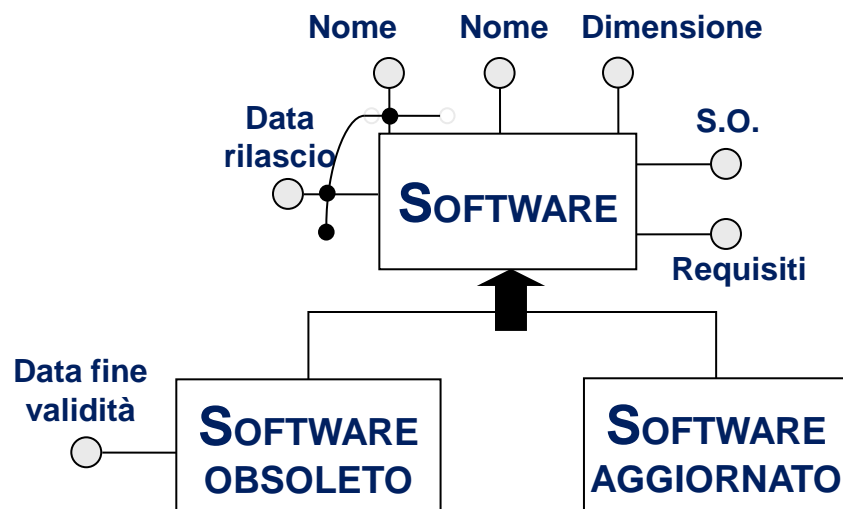
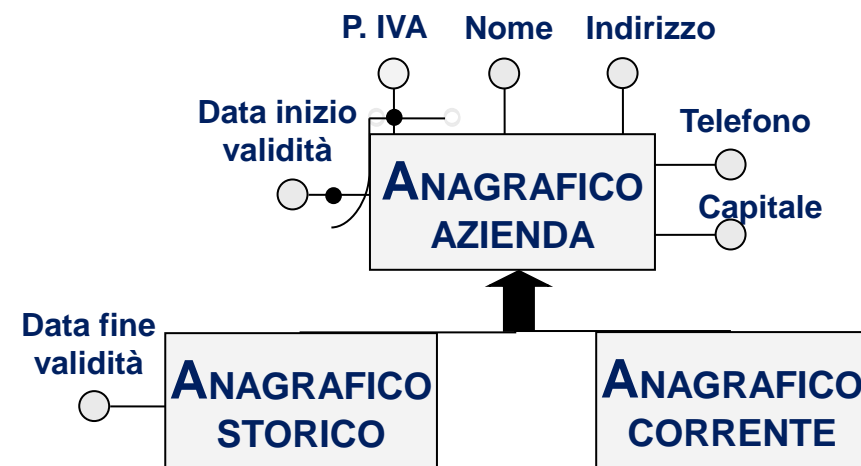


Vincolo non esprimibile in E-R. Scriverlo in linguaggio naturale come vincolo di integrità. Verrà risolto tramite trigger durante la progettazione fisica.

# Storicizzazione di concetto

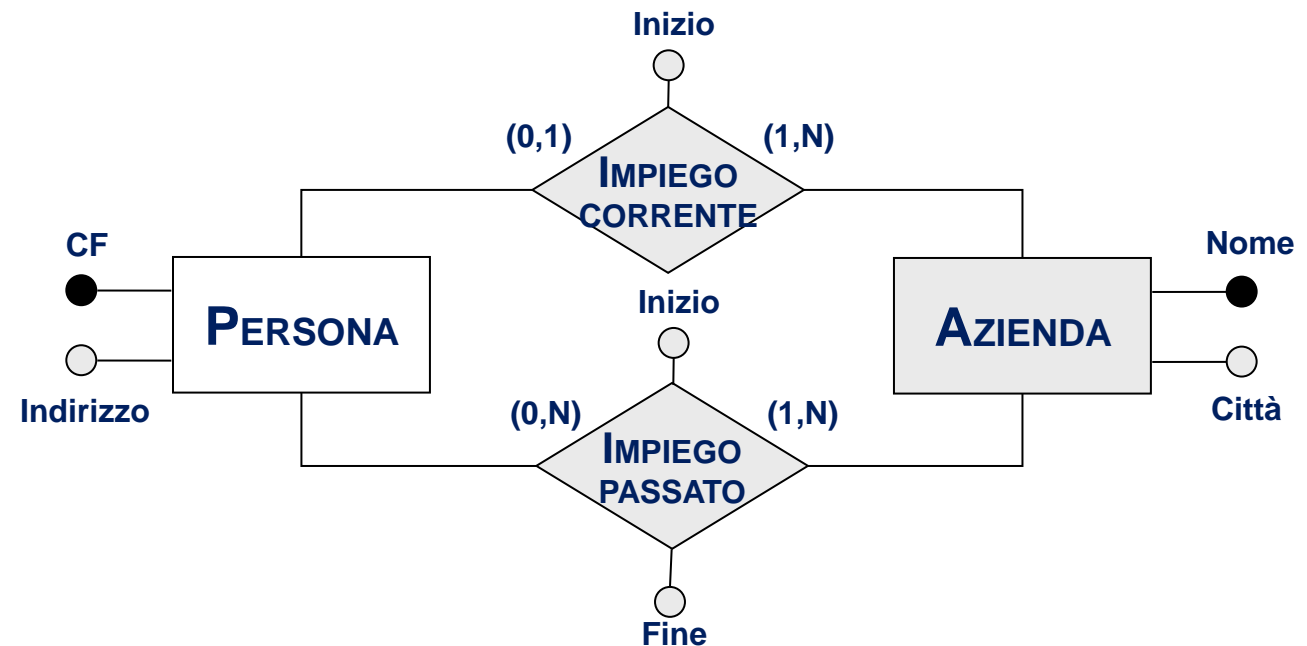
Vogliamo memorizzare le informazioni correnti di un'azienda, tenendo però traccia dei dati che sono variati.

Soluzione efficace: uso 2 entità con gli stessi attributi, una rappresenta le informazioni aggiornate, l'altra lo storico.

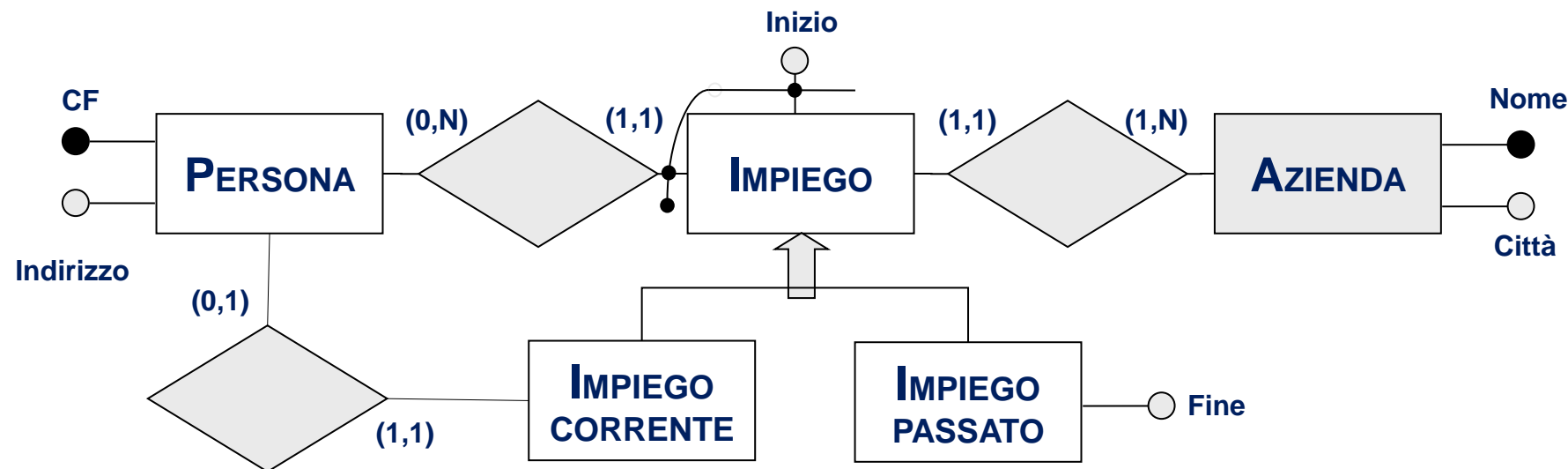


# Storicizzazione di concetto (relazione)

- Si vuole storicizzare un concetto rappresentato da una relazione.
- Possibile soluzione: quella a destra. Notare le differenti cardinalità delle partecipazioni dell'entità Persona alle due relazioni.
- Svantaggi: non possiamo rappresentare il fatto che una persona possa aver lavorato, in periodi differenti, per la stessa azienda. *(altrimenti avremmo due occorrenze identiche nella relazione Impiego Passato)*
- **Soluzione: reificazione delle relazioni.**

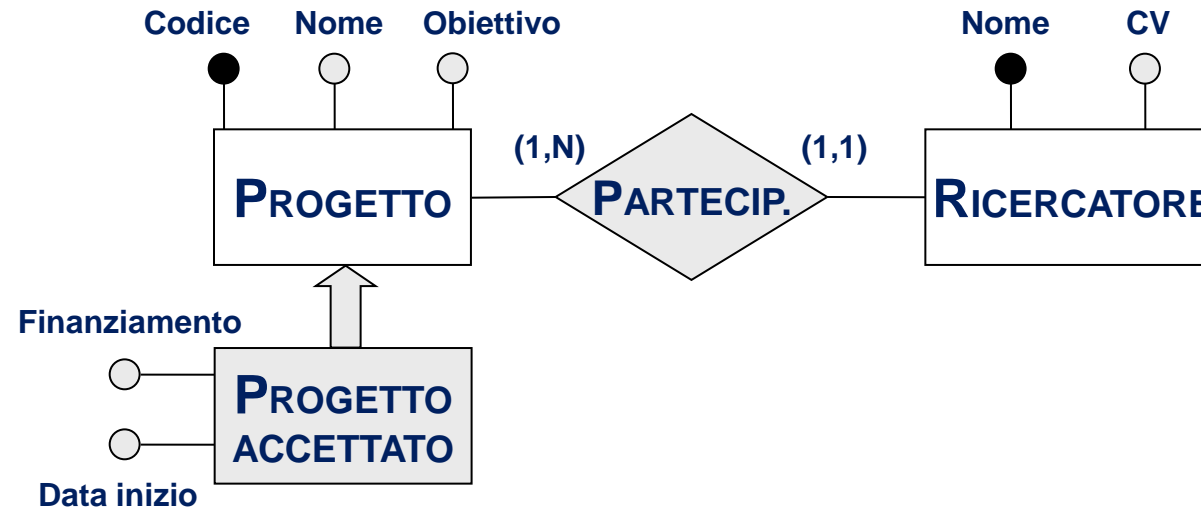


## Storicizzazione di concetto (relazione)



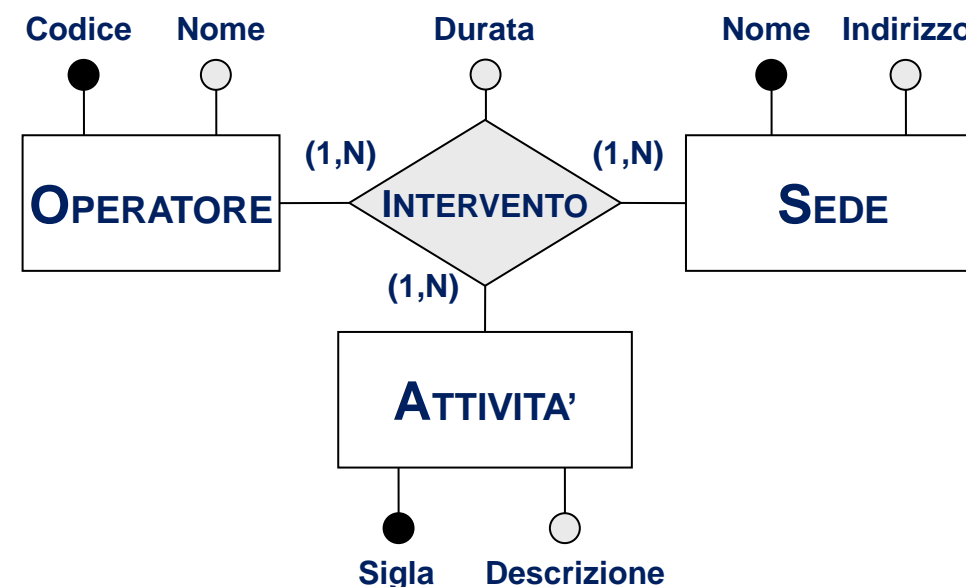
- Usiamo una generalizzazione per gestire la storicizzazione di Impiego.
- Attenzione: anche in questo caso risulta necessario l'inserimento di un vincolo esterno allo schema che impone che tutte le occorrenze della relazione tra *Persona* e *Impiego Corrente* compaiano anche tra le occorrenze della relazione tra *Persona* e *Impiego*.

## Evoluzione di concetto



# Relazione ternaria

- Negli schemi E-R, le relazioni ternarie si incontrano raramente.
- Le relazioni che coinvolgono più di 3 entità sono fortemente *sconsigliate*, perchè tipicamente cercano di rappresentare con un unico costrutto entità tra loro indipendenti.
- Nell'esempio a destra, l'uso della relazione ternaria risulta una buona soluzione nel caso in cui:
  - 1) Un operatore può effettuare operazioni che consistono in attività diverse svolte in sedi diverse
  - 2) In ogni sede, possono operare operatori diversi svolgendo attività diverse
  - 3) Le attività possono essere svolte da operatori diversi e in sedi diverse
- Se la realtà da modellare non è questa (e.g. ci sono vincoli più stretti sui punti 1,2,3), allora la reificazione di Intervento è necessaria.

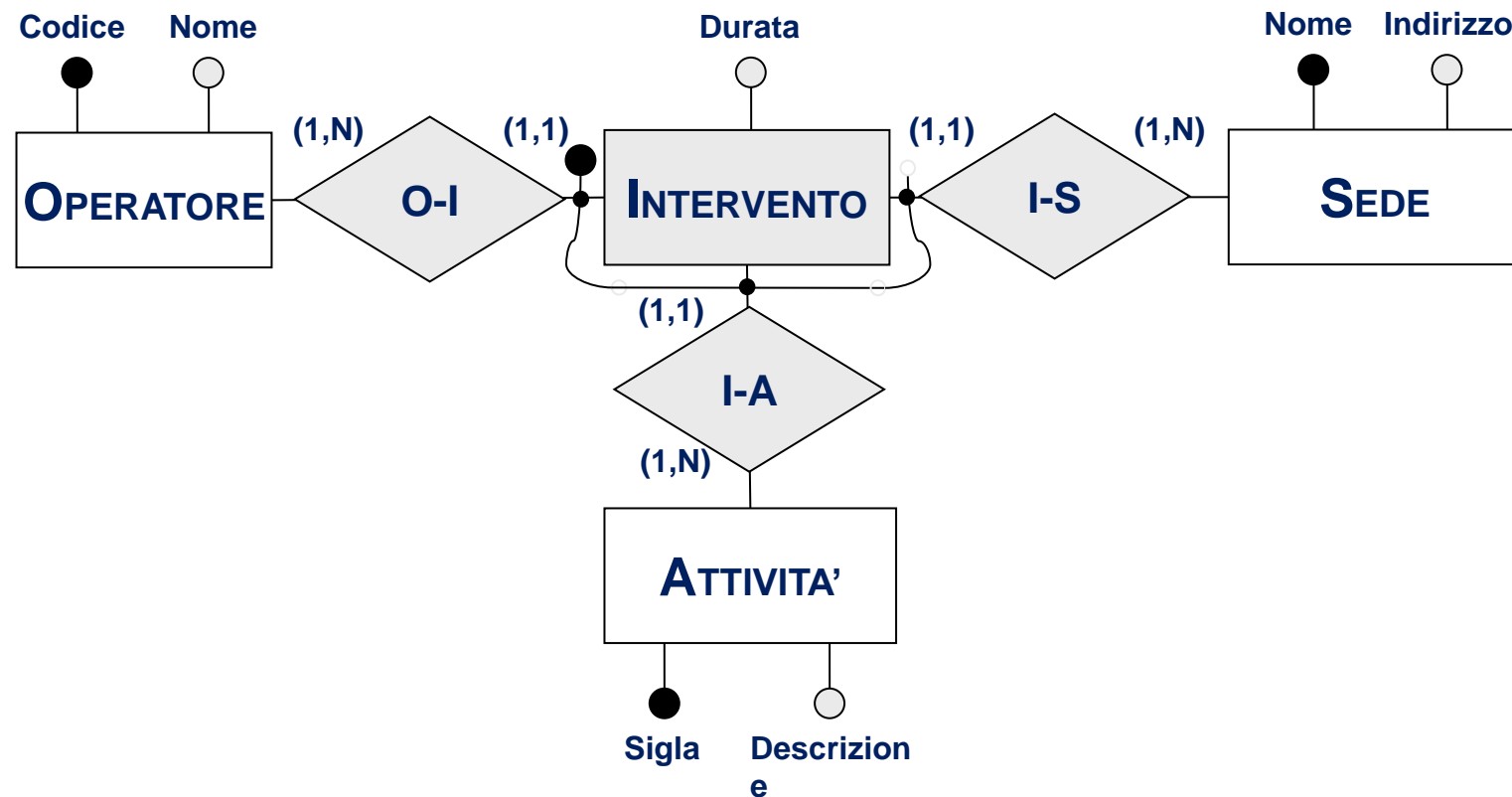




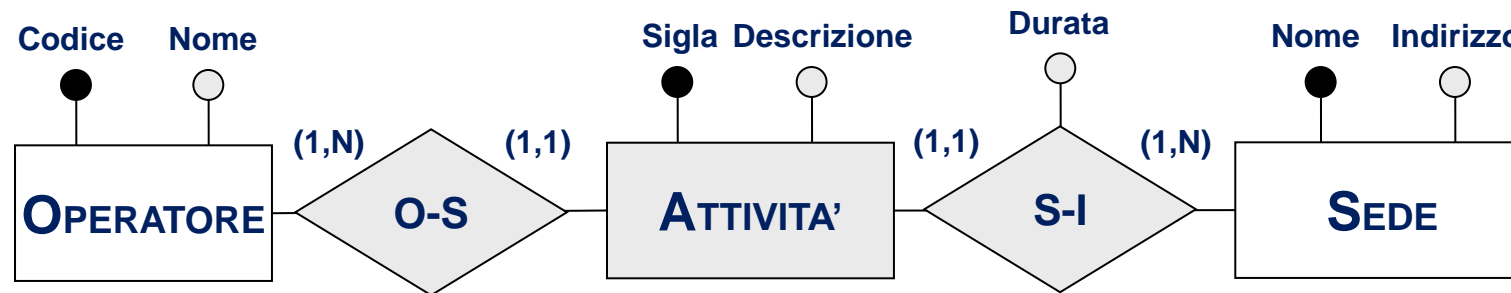
# Reificazione di relazione ternaria



- Il nuovo schema modella esattamente la situazione dello schema originario perchè *la nuova entità risulta identificata da tutte le entità originarie*.
- Cambiando opportunamente l'identificazione, siamo però in grado di modellare con questo pattern altre situazioni per le quali la relazione ternaria non sarebbe corretta.
- Esempio:
  - “in ogni sede ogni operatore svolge sempre la stessa attività”: identificatore Sede-Operatore.
  - “in ogni sede ogni attività viene svolta sempre dallo stesso operatore”: identificatore Attività-Sede.
  - “ogni operatore svolge ogni attività in una sola sede”: identificatore Operatore-Attività.



## Reificazione di relazione ternaria 2



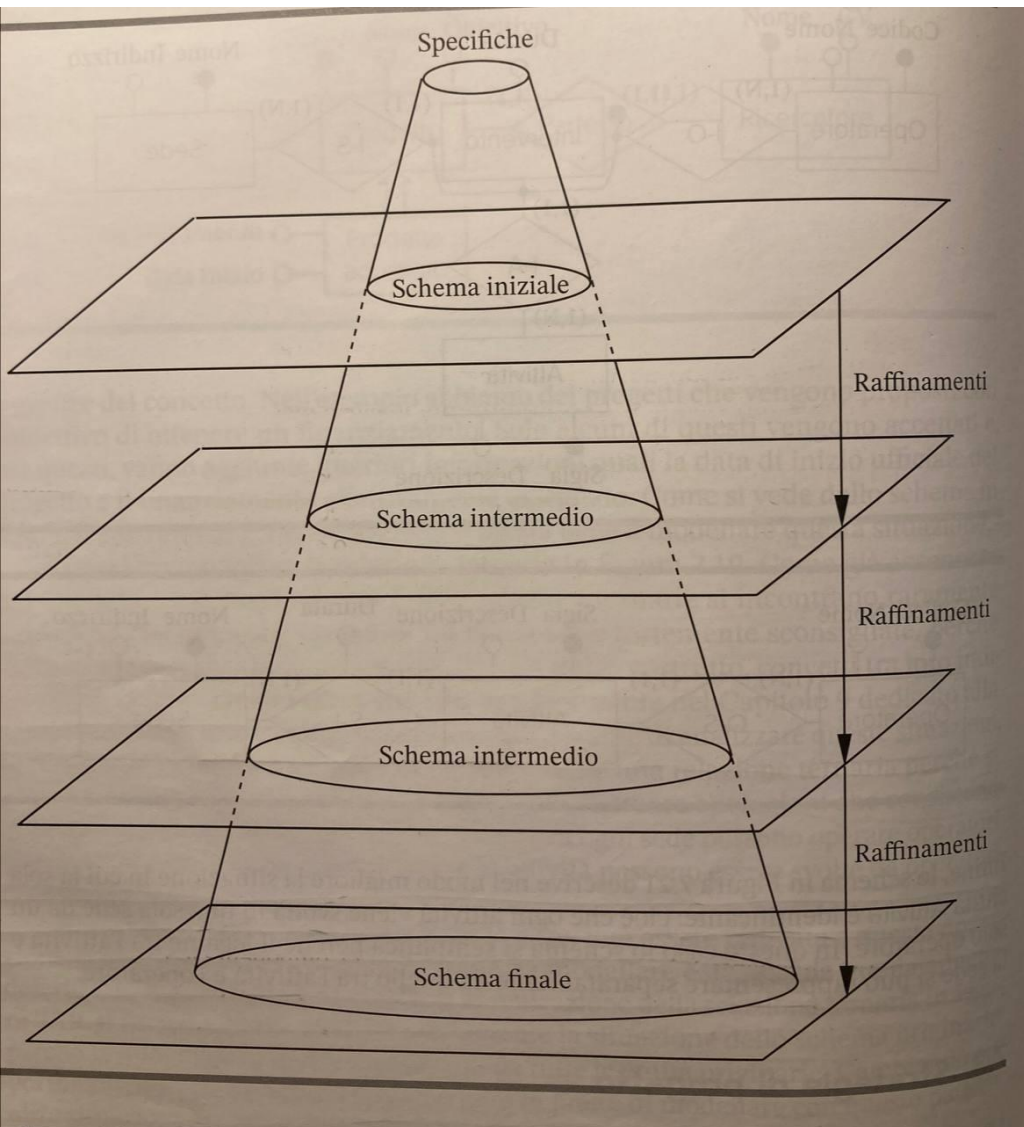
- Ogni attività viene svolta in una sola sede da un solo operatore.
- Lo schema, in questo caso, si semplifica perchè il legame tra l'attività e la sede si può rappresentare separatamente da quello tra l'attività e l'operatore.

## Strategie di progetto

- Lo sviluppo di uno *schema concettuale* a partire dalle sue *specifiche* può essere considerato a tutti gli effetti un processo di ingegnerizzazione.
- Strategie:
  - top-down
  - bottom-up
  - inside-out



## Strategia top-down



- Lo schema concettuale viene prodotto mediante una serie di *raffinamenti* successivi, a partire da uno *schema iniziale* che descrive tutte le specifiche con pochi concetti molto astratti.
- Lo schema viene poi raffinato mediante opportune trasformazioni (chiamate primitive di trasformazione top-down) che aumentano il dettaglio dei vari concetti.
- Ogni livello (vedi figura a sinistra) descrive le medesime informazioni a un diverso livello di dettaglio: *tutte le informazioni presenti nello schema finale sono presenti a ogni livello di raffinamento.*

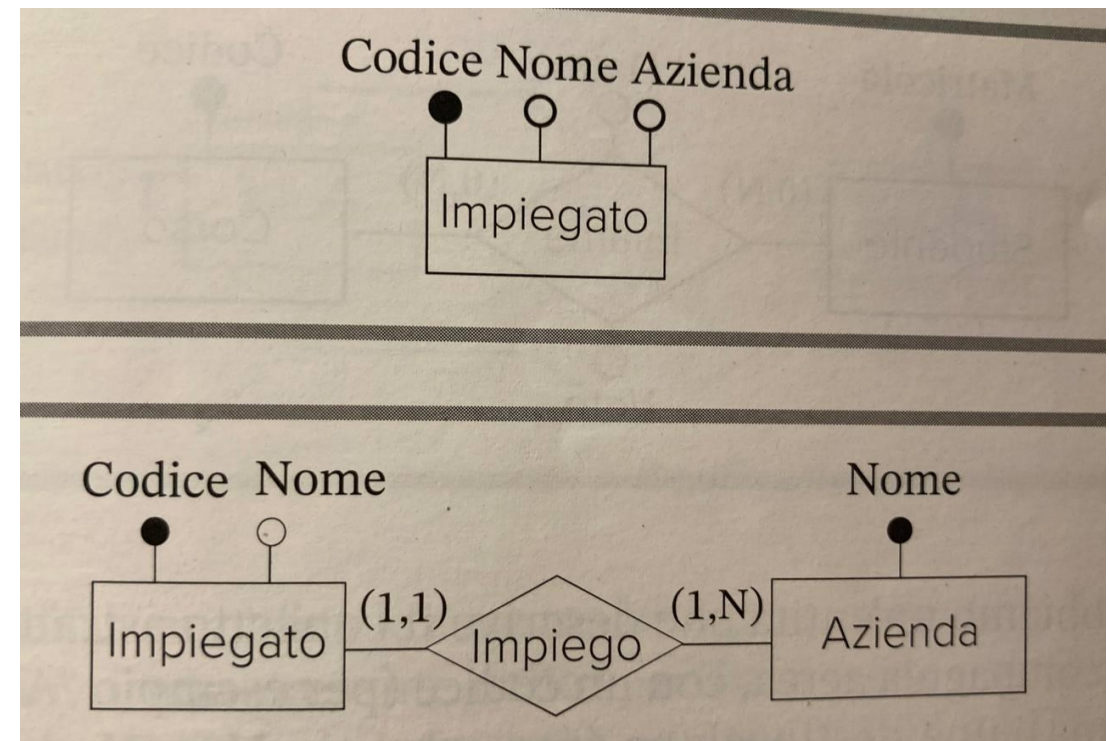
## Primitive di raffinamento top-down

1) Definizione degli attributi di un'entità o di una relazione. E.g.: la specifica, per una entità Persona, di tutti gli attributi di interesse:

- Codice Fiscale
- Cognome
- Età
- Sesso
- Città di nascita

2) Reificazione di un attributo o di un'entità.

- Esempio: figura a destra





## Primitive di raffinamento top-down

3) Decomposizione di una relazione in due relazioni distinte.

impiego corrente

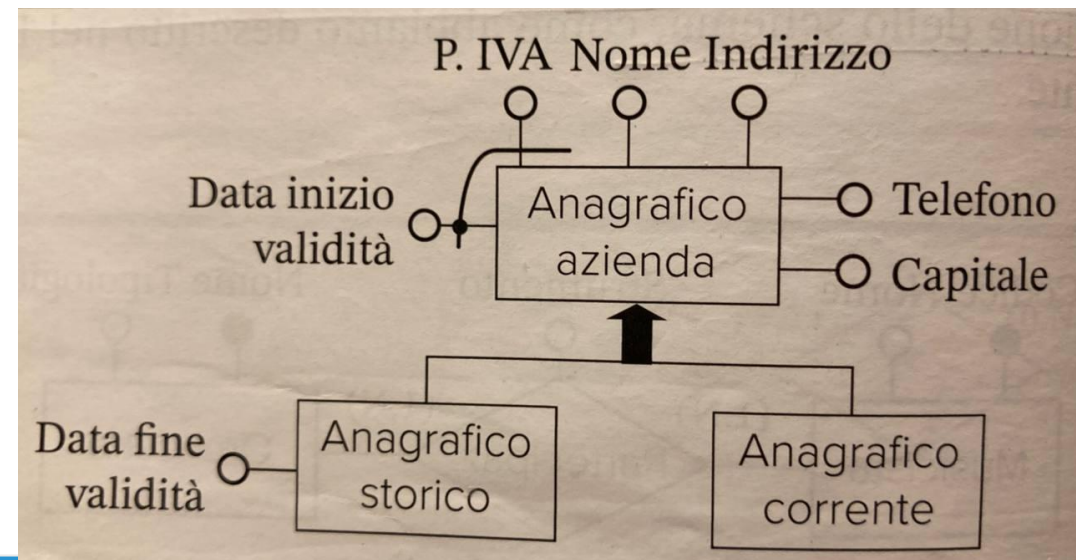
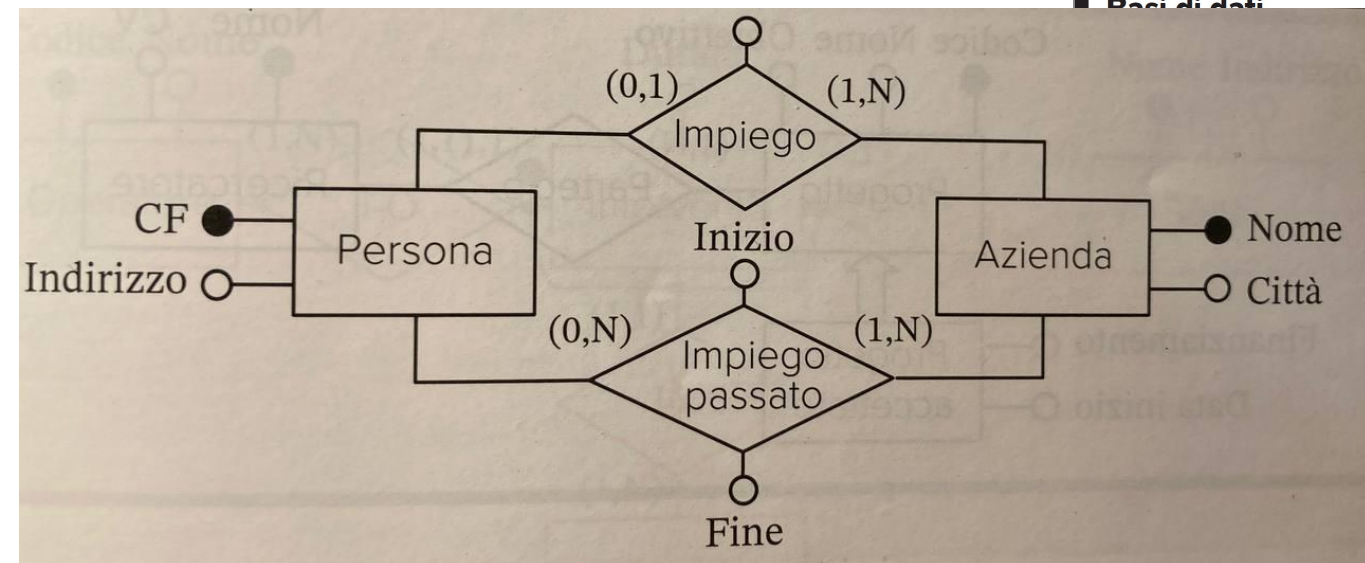
vs

impiego passato

3) Trasformazione di un'entità in una gerarchia di generalizzazione:

anagrafico azienda

- anagrafico storico
- anagrafico corrente



# Vantaggi e svantaggi dell'approccio top-down



### Vantaggi:

- Il progettista può descrivere inizialmente tutte le specifiche dei dati trascurandone i dettagli
- E solo dopo entrare nel merito di un concetto alla volta

### Svantaggi:

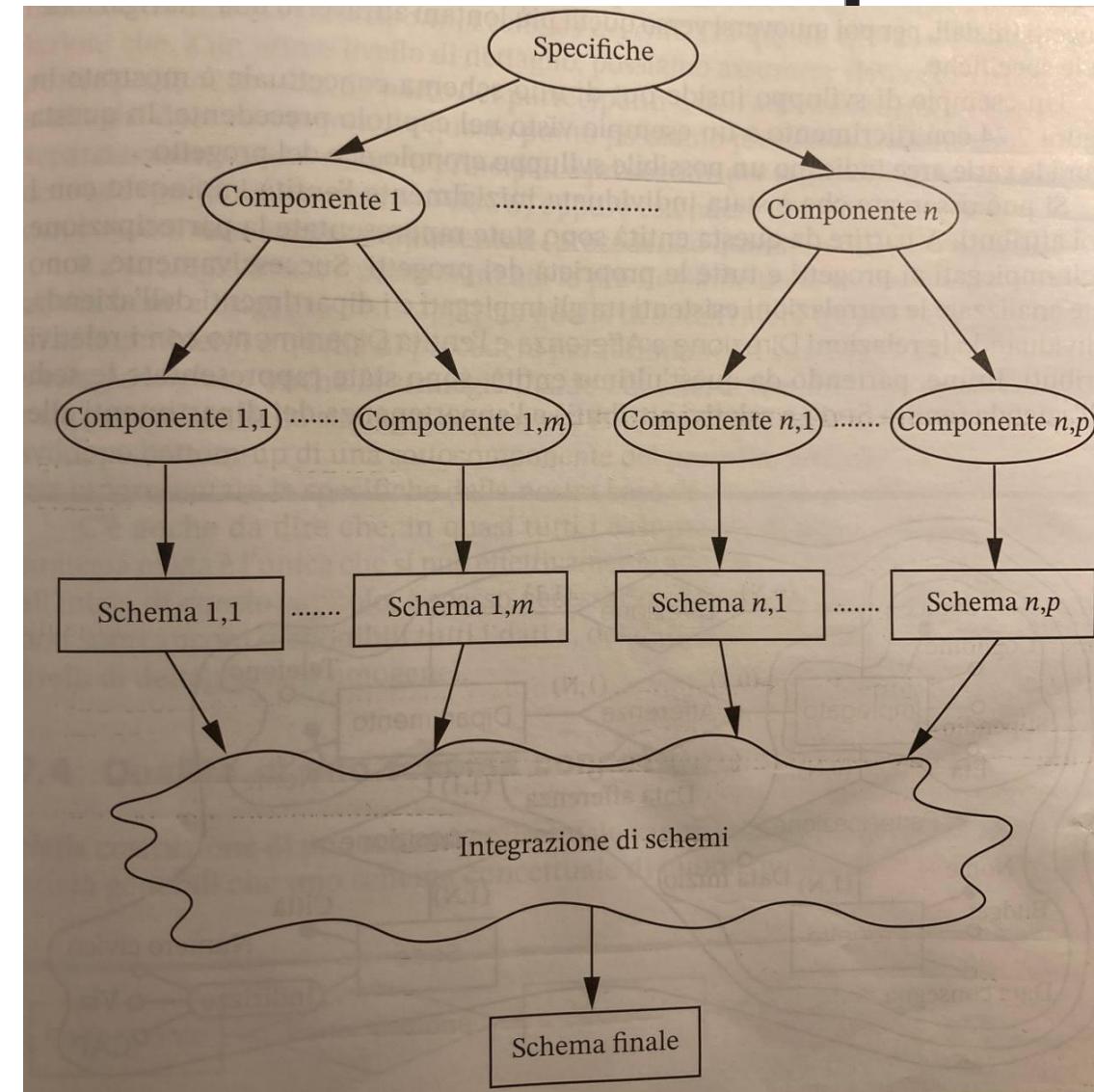
- È possibile solo quando si possiede, ***sin dall'inizio***, una visione globale e astratta di *tutte* le componenti del sistema.
  - Può risultare difficile con applicazione complesse.



# Strategia bottom-up



- Le specifiche iniziali sono suddivise in componenti via via sempre più piccole *fino a quando* tali componenti descrivono un frammento elementare della realtà di interesse.
- A questo punto, le varie componenti vengono rappresentate da *semplici schemi concettuali* (che possono consistere anche in singoli concetti).
- Successivamente, gli schemi vengono **integrati**, fino a giungere allo schema finale.
- Anche in questo caso, lo schema finale si ottiene tramite alcune primitive di trasformazione bottom-up:
  - E.g: aggregazione di un insieme di attributi in un'entità:
  - Integrazione di “matricola”, “Cognome”, “Data di nascita”, “città di nascita” nell'entità “Studente”.

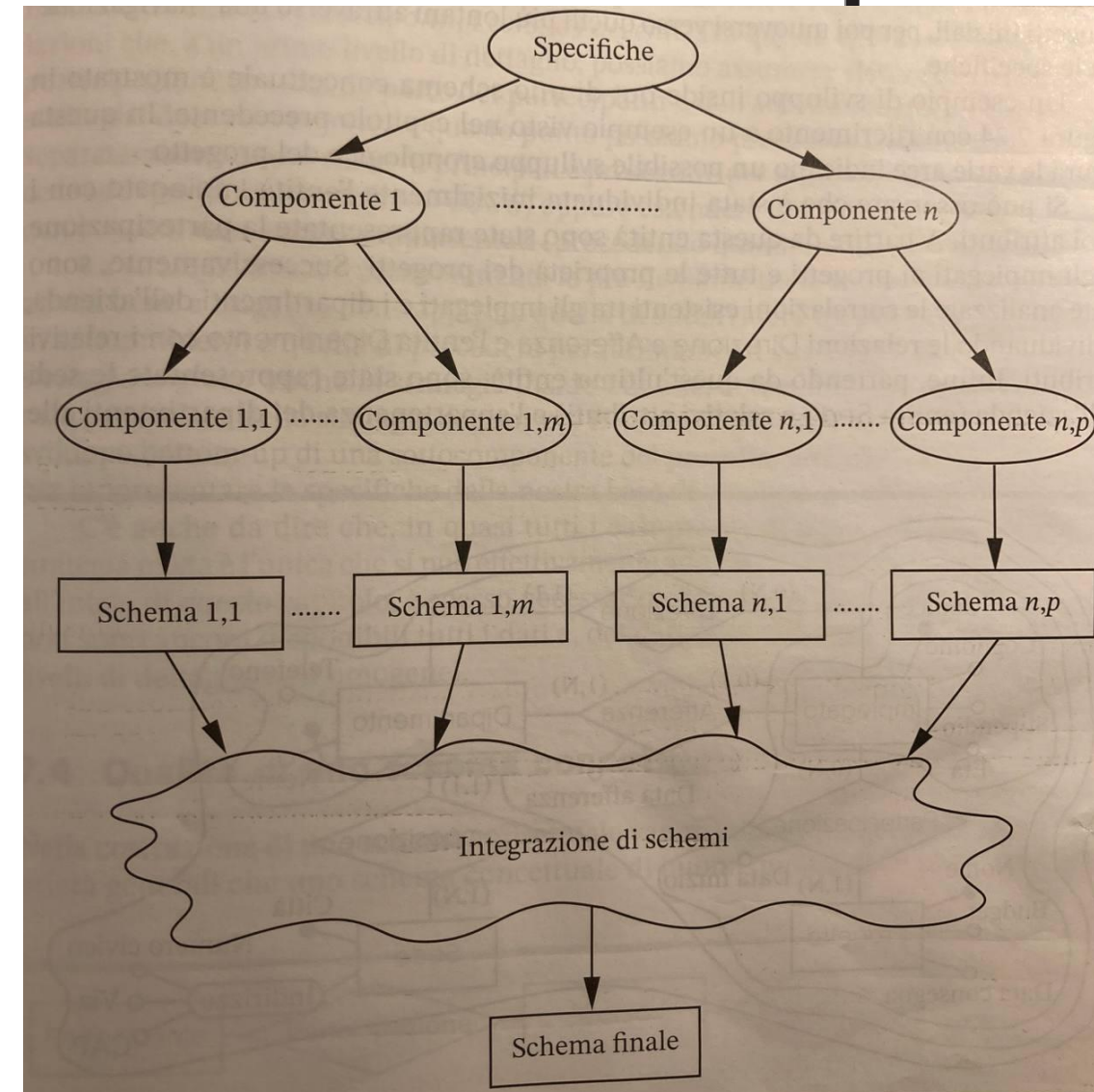


## Strategia bottom-up

- Vantaggi:
- Si adatta molto bene ad una decomposizione del problema (**divisione dei compiti**).

### Svantaggi:

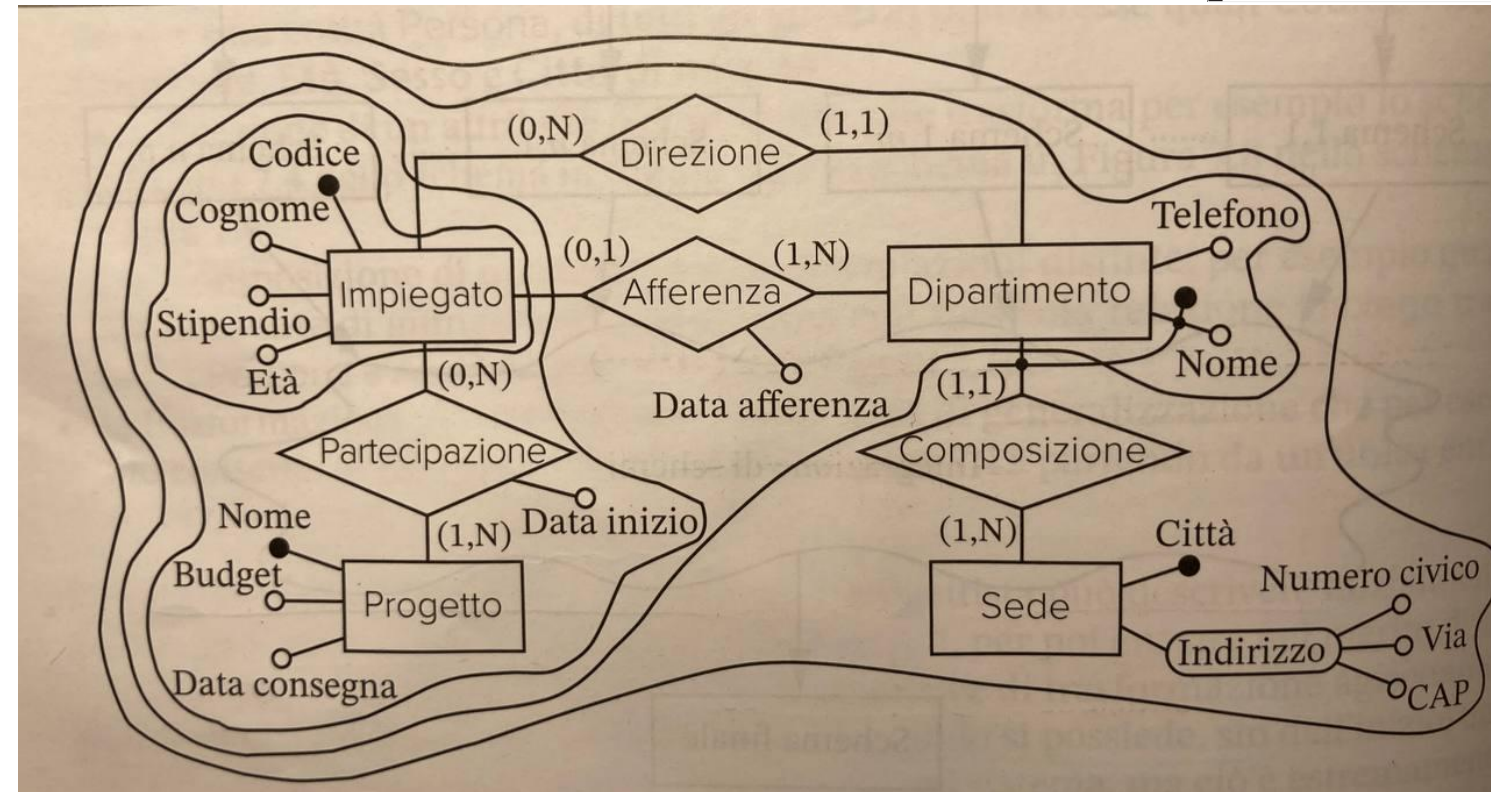
- Richiede operazioni di integrazione.
- Nel caso di schemi concettuali complessi, questo può essere un problema.





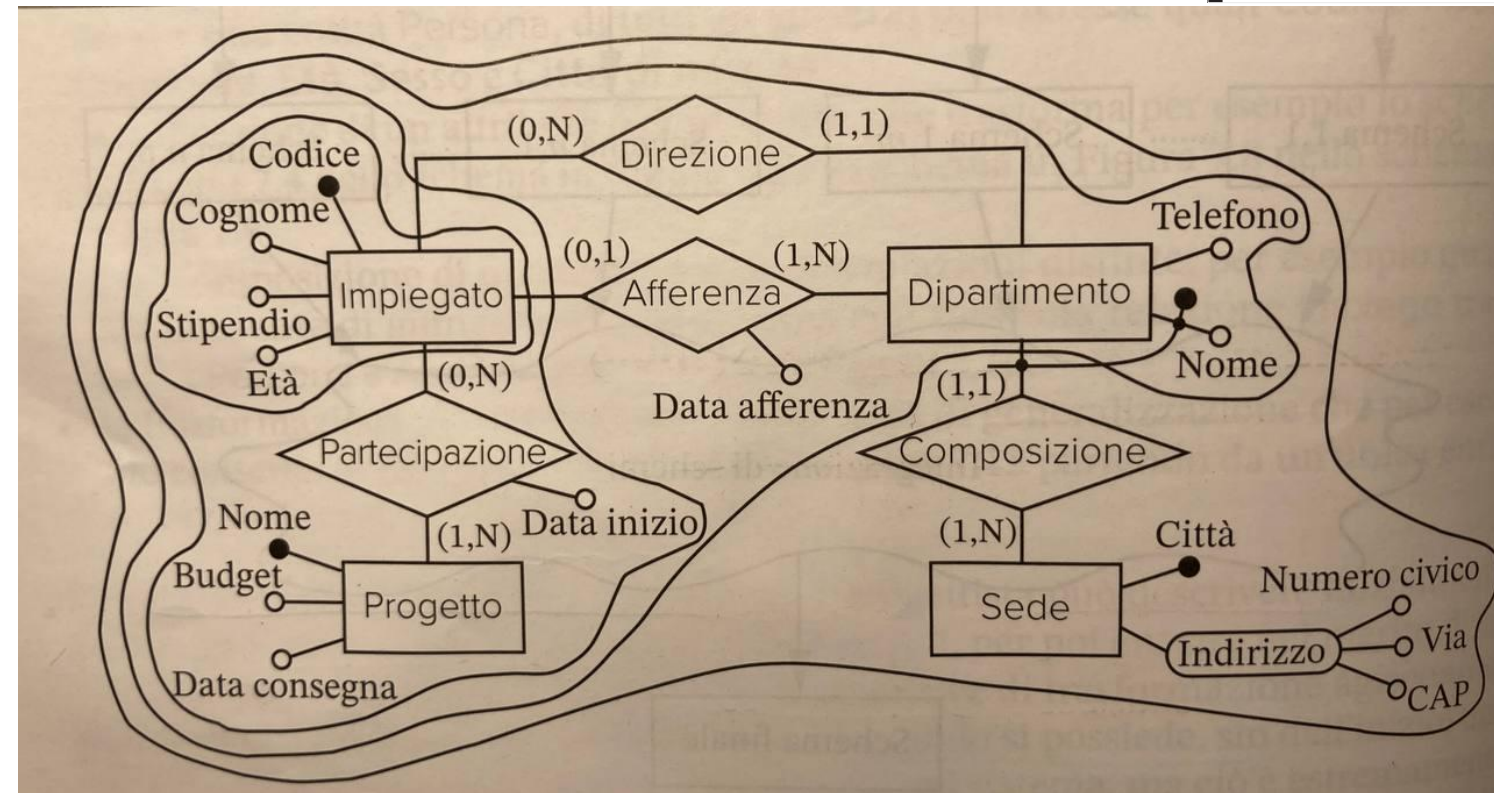
# Strategia inside-out

- Si individuano inizialmente solo alcuni concetti importanti (vedi Impiegato).
- A partire da questi, si procede a **macchia d'olio**. Ovvero:
  - Si rappresentano prima i concetti in relazione con i concetti iniziali
  - Per poi muoversi verso quelli più lontani attraverso una navigazione tra i requisiti.



## Strategia inside-out

- **FASE 1:** è stata individuata l'entità Impiegato, con i suoi attributi.
- **FASE 2:** a partire da Impiegato, sono stati aggiunti la partecipazione degli impiegati ai progetti e tutte le proprietà dei progetti.
- **FASE 3:** poi, vengono analizzate le correlazioni esistenti tra gli impiegati e i dipartimenti dell'azienda.
  - **Nota:** in questa fase, non è ancora possibile scegliere un identificatore per l'entità Dipartimento, perché è possibile avere dipartimenti con lo stesso nome in sedi diverse.
- **FASE 4:** partendo da Dipartimento, sono state rappresentate le sedi dell'azienda e l'appartenenza dei dipartimenti alle relative sedi.
  - **Nota:** ora si può procedere all'identificazione dell'entità Dipartimento.





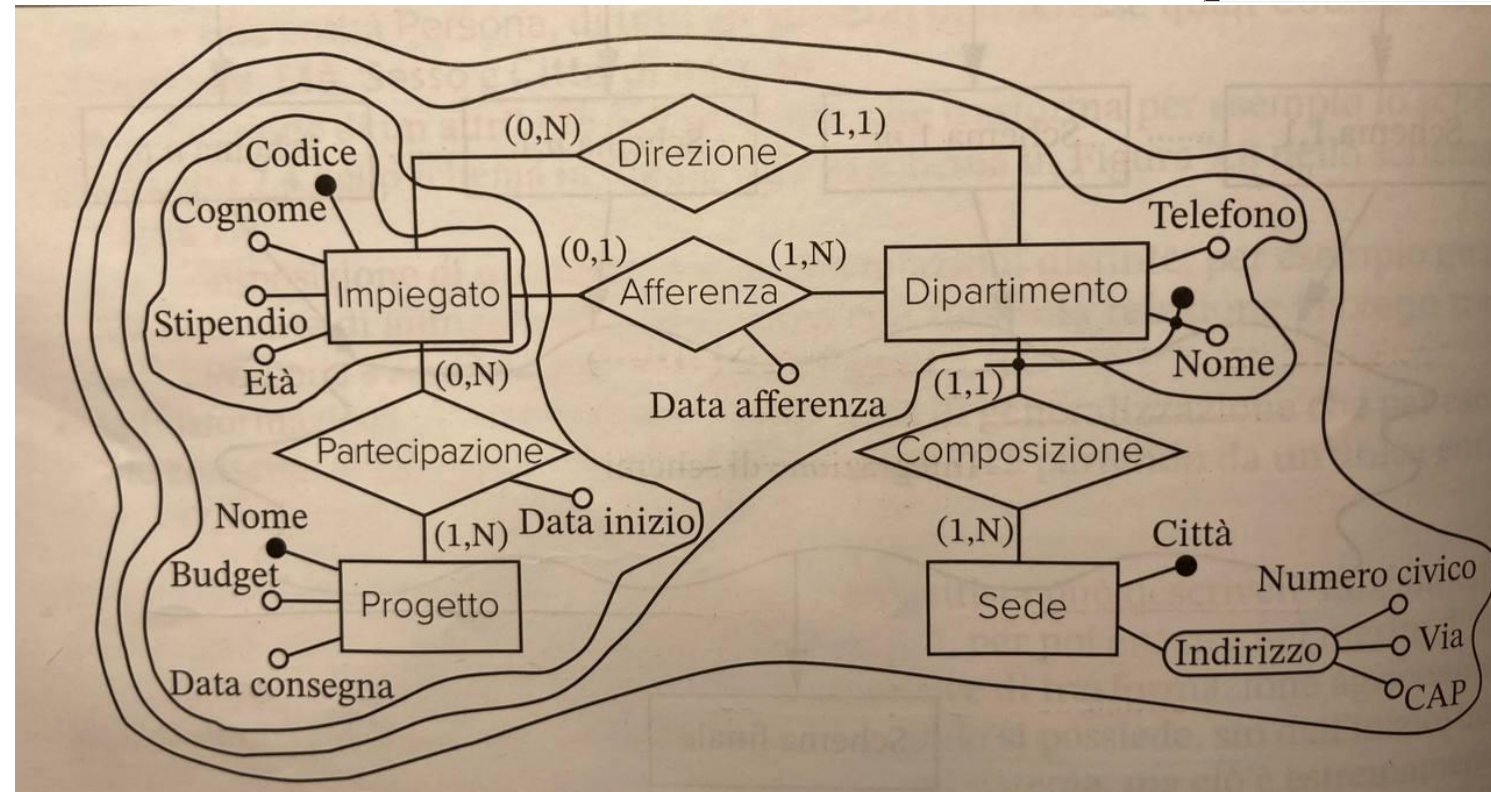
## Strategia inside-out

### Vantaggi:

- Non richiede passi di integrazione

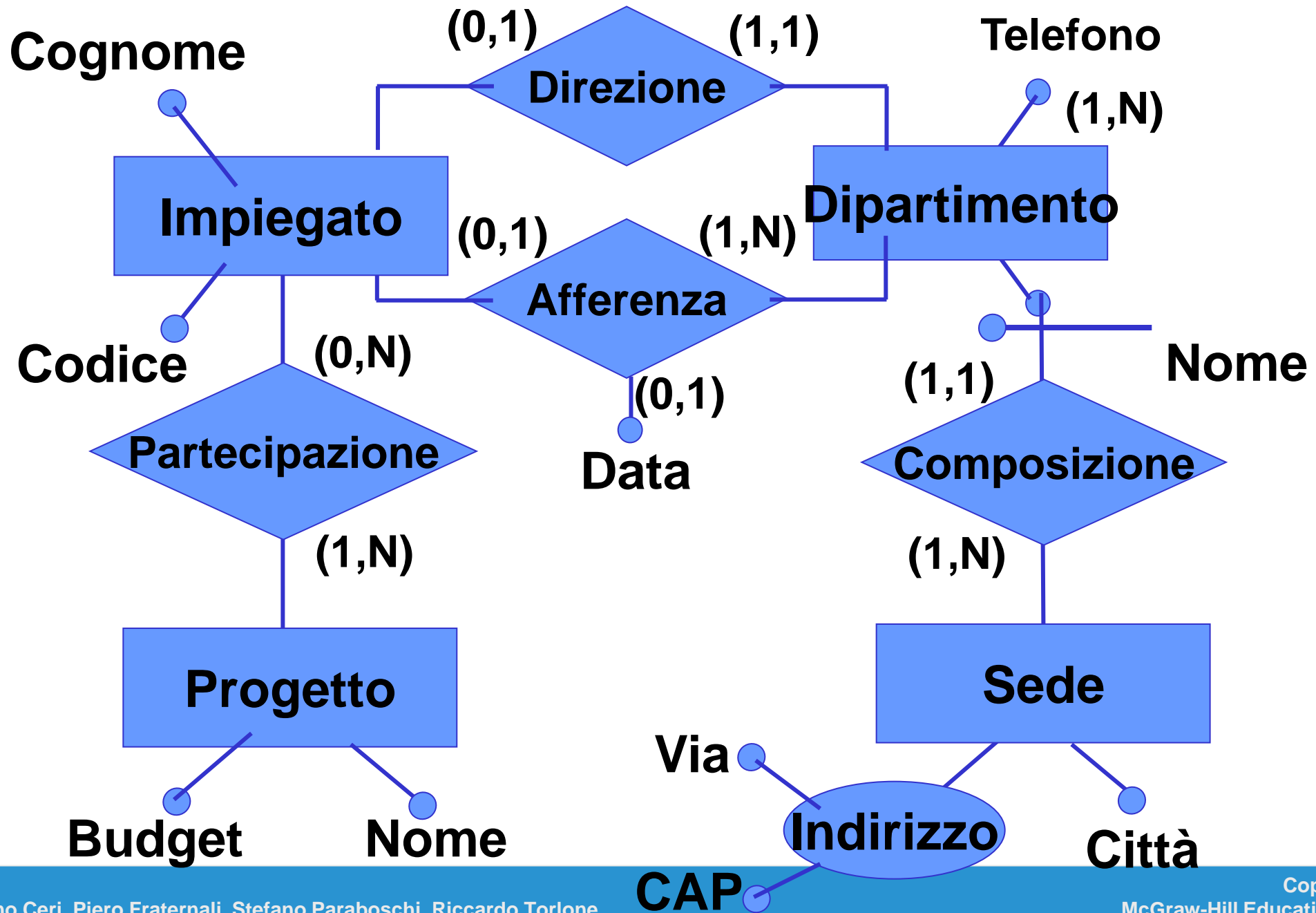
### Svantaggi:

- Non è possibile procedere per livelli di astrazione.





## Strategia inside-out: un esempio



## Strategia mista

- Si procede di solito con una strategia **mista**:
  - Si cerca di combinare i vantaggi delle strategie top-down e bottom-up
  - Si individuano i concetti principali e si realizza uno **schema scheletro**
    - Fornisce una visione unitaria dell'intero progetto
  - Sulla base di questo si può decomporre
  - Poi si raffina, si espande, si integra

**Nota:** nel progetto potete scrivere quale strategia avete usato.





# Qualità di uno schema concettuale

- 1) Correttezza: uno schema concettuale è *corretto* quando utilizza propriamente i costrutti messi a disposizione dal modello concettuale di riferimento.
  - Errori *sintattici*: riguardano un uso non ammesso di costrutti, e.g. una generalizzazione tra relazione invece che tra entità.
  - Errori *semantici*: riguardano un uso di costrutti che non rispetta la loro definizione, e.g. l'uso di una relazione per descrivere il fatto che un'entità è specializzazione di un'altra.
- 2) Completezza: uno schema concettuale è *completo* quando rappresenta tutti i dati di interesse e quando tutte le operazioni possono essere eseguite a partire dai concetti descritti nello schema.
  - La completezza di uno schema si può verificare controllando che *tutte le specifiche* sui dati siano rappresentate da qualche concetto presente nello schema.



# Qualità di uno schema concettuale

### 3) Leggibilità

- 4) Minimalità: uno schema è *minimale* quando tutte le specifiche sui dati sono rappresentate una sola volta. Quindi, uno schema non è minimale quando esistono delle ridondanze, ovvero concetti che possono essere derivati da altri.

Possibili fonti di ridondanza:

- Presenza di cicli dovuti a relazioni e/o specializzazioni
- Attributi derivati

Non sempre una ridondanza è indesiderata: a volte, può nascere da precise scelte progettuali.

La scelta tra il *mantenere* o l'*eliminare* le ridondanze è compito della fase dell'*analisi delle ridondanze*, sotto fase della progettazione logica.

Ne riparleremo fra qualche lezione.



## Una metodologia

- **Analisi dei requisiti**
  - Analizzare i requisiti ed eliminare le ambiguità
  - Costruire un glossario dei termini
  - Raggruppare i requisiti in insiemi omogenei
- **Passo base**
  - Definire uno schema scheletro con i concetti più rilevanti
- **Passo iterativo**  
(da ripetere finché non si è soddisfatti)
  - Raffinare i concetti presenti sulla base delle loro specifiche
  - Aggiungere concetti per descrivere specifiche non descritte
- **Analisi di qualità**  
(ripetuta e distribuita)
  - Verificare le qualità dello schema e modificarlo

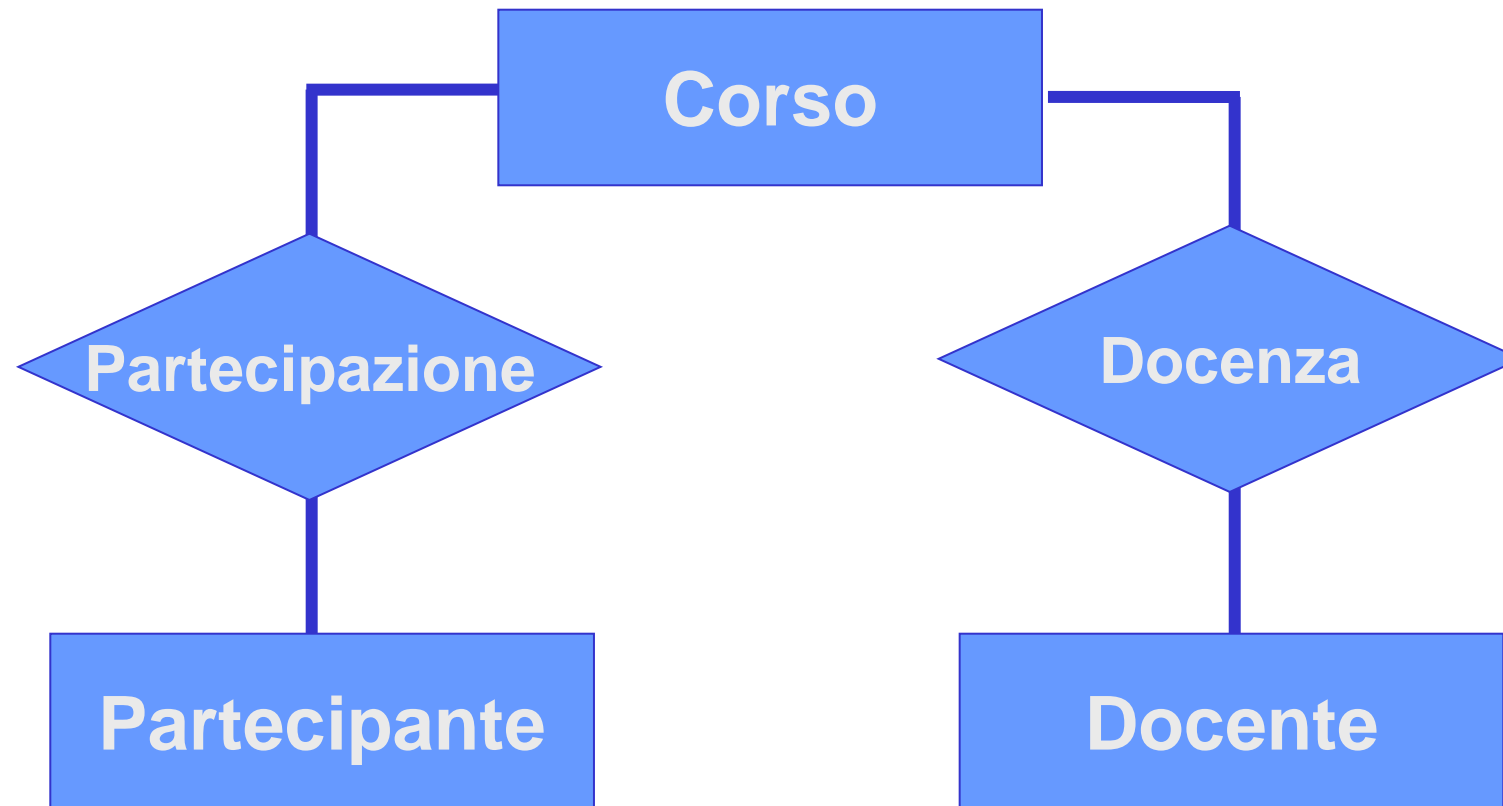


# Un esempio di progettazione concettuale (società di formazione)

## **Frasi di carattere generale**

**Si vuole realizzare una base di dati per una società che eroga corsi, di cui vogliamo rappresentare i dati dei partecipanti ai corsi e dei docenti.**

## Schema scheletro



# Schema scheletro

A questo punto, decidiamo di analizzare *separatamente* le specifiche riguardanti i partecipanti, quelle riguardanti i corsi, e quelle riguardanti i docenti.

Procediamo a *macchia d'olio* (strategia inside-out) per includere tutti i concetti non presi in considerazione dallo schema scheletro.

Incontreremo in molti casi i *pattern di progetto* che abbiamo discusso nelle lezioni precedenti.



## **Fraasi relative ai partecipanti**

**Per i partecipanti (circa 5000), identificati da un codice, rappresentiamo il codice fiscale, il cognome, l'età, il sesso, la città di nascita, i nomi dei loro attuali datori di lavoro e di quelli precedenti (insieme alle date di inizio e fine rapporto), le edizioni dei corsi che stanno attualmente frequentando e quelli che hanno frequentato nel passato, con la relativa votazione finale in decimi.**



### **Fraasi relative ai datori di lavoro**

**Relativamente ai datori di lavoro presenti e passati dei partecipanti, rappresentiamo il nome, l'indirizzo e il numero di telefono.**

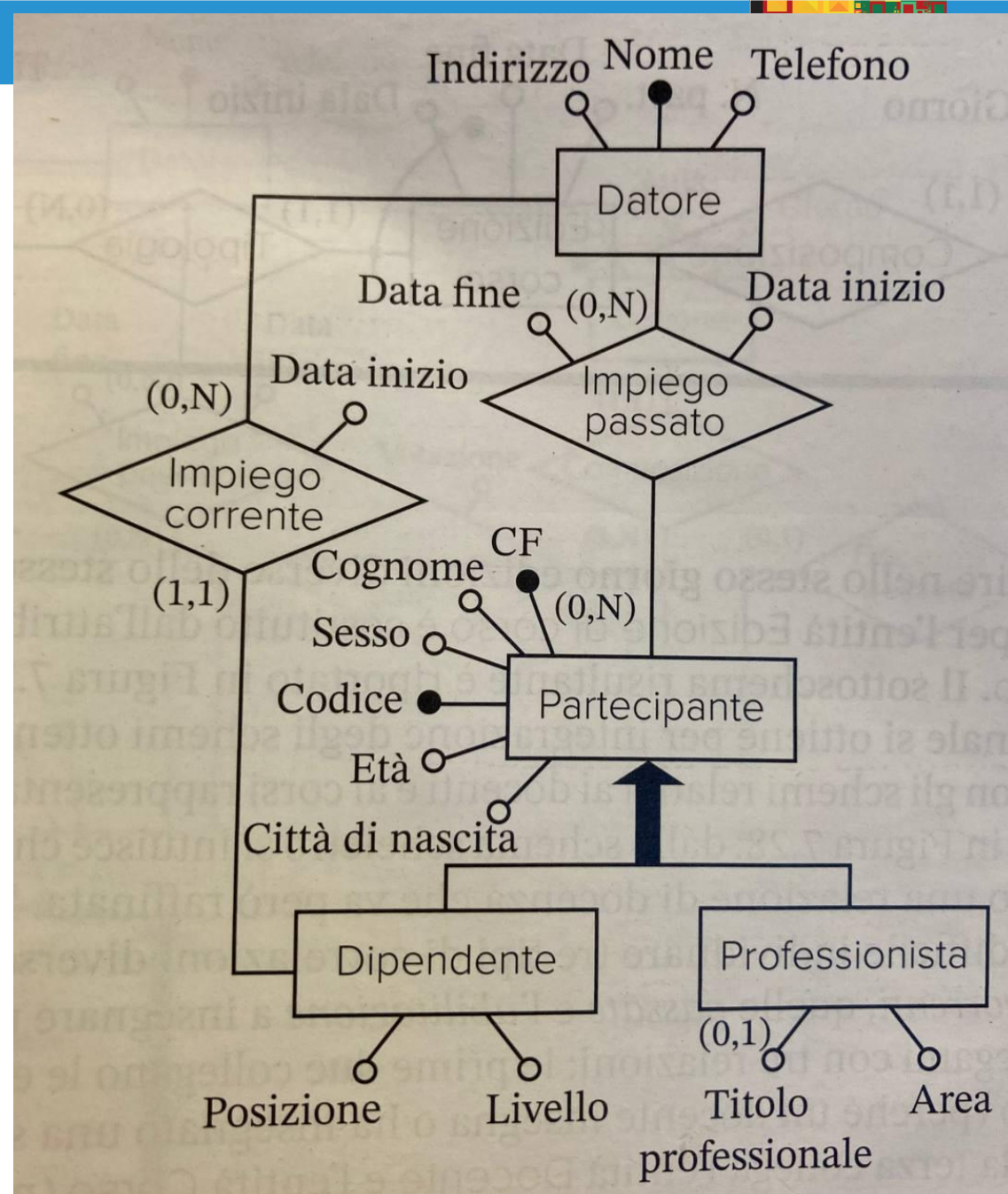
### **Fraasi relative a tipi specifici di partecipanti**

**Per i partecipanti che sono liberi professionisti, rappresentiamo l'area di interesse e, se lo possiedono, il titolo professionale. Per i partecipanti che sono dipendenti, rappresentiamo invece il loro livello e la posizione ricoperta.**



- Si individuano facilmente 2 tipologie di partecipanti:
  - 1) Professionisti
  - 2) Dipendenti
- Questi concetti sono rappresentabili come entità figlie dell'entità *Partecipante*: la generalizzazione che ne risulta è *totale*.
- Ora vanno rappresentati gli impieghi dei partecipanti.
  - Introduciamo l'entità *Datore*.
  - Vanno rappresentati due concetti distinti:
    - 1) *Impiego passato*
      - Attributi "Data Fine" e "Data Inizio"
      - Lega i Partecipanti ai Datori
    - 2) *Impiego presente*
      - Attributi "Data Inizio"
      - Lega i Dipendenti ai Datori
  - Aggiungiamo ora i restanti attributi
  - ... e i vincoli di partecipazione

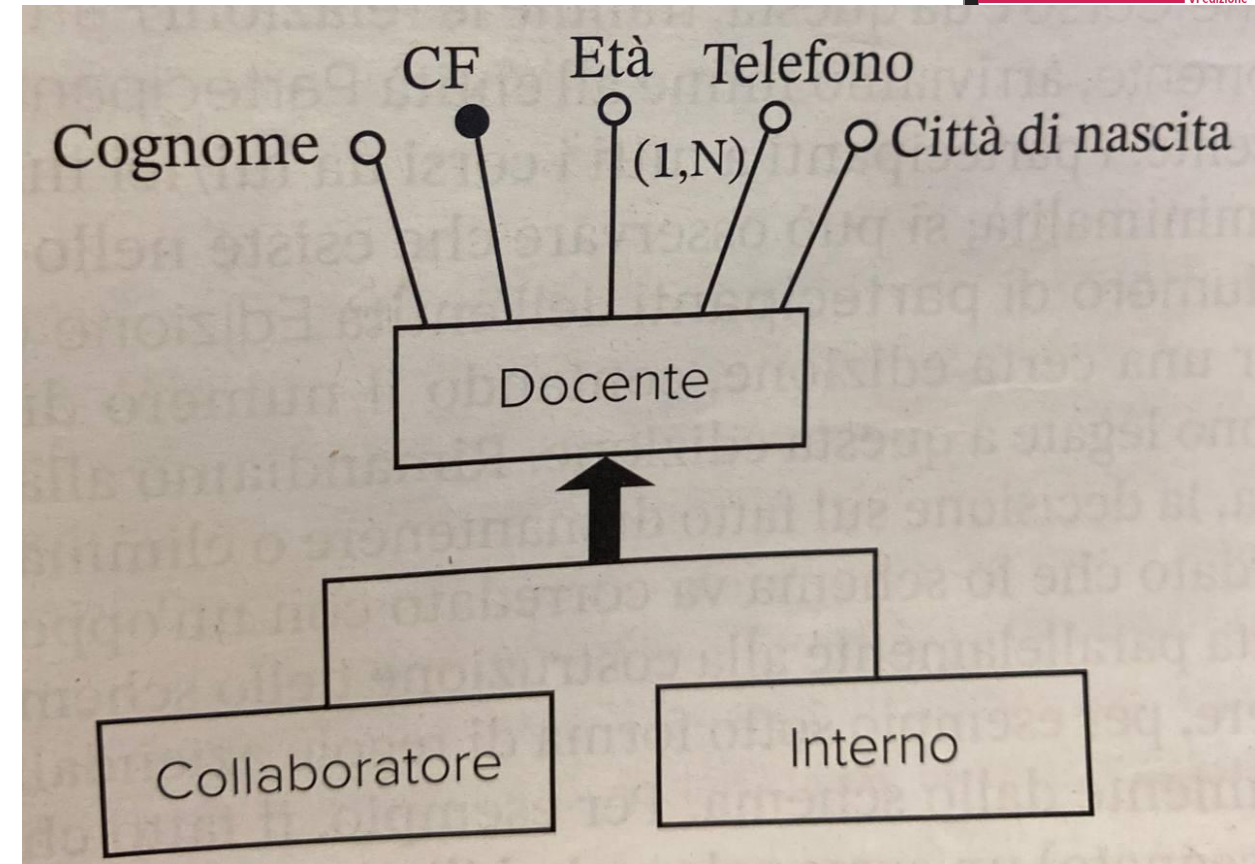
**Nota:** l'entità *Partecipante* ha due identificatori: il *codice interno* dell'azienda e il *codice fiscale*. Inoltre, l'attributo "Titolo professionale" è opzionale.



### Frasi relative ai docenti

**Per i docenti (circa 300), rappresentiamo il cognome, l'età, la città di nascita, tutti i numeri di telefono, il titolo del corso che insegnano, di quelli che hanno insegnato nel passato e di quelli che possono insegnare. I docenti possono essere dipendenti interni della società di formazione o collaboratori esterni.**

- Docenti:
- Vanno distinti in:
  - 1) Dipendenti interni
  - 2) Collaboratori esterni
  - 3)--> Generalizzazione totale.
- Aggiungiamo gli attributi
  - “cognome”
  - “età”
  - “città di nascita”
  - “numero di telefono” : *multivalore*
- Osserviamo che questi attributi **non** forniscono nessun identificatore naturale per l'entità *Docente*.
  - Ne cerchiamo uno al di fuori delle specifiche
  - Il “codice fiscale” (CF) risulta una buona scelta.

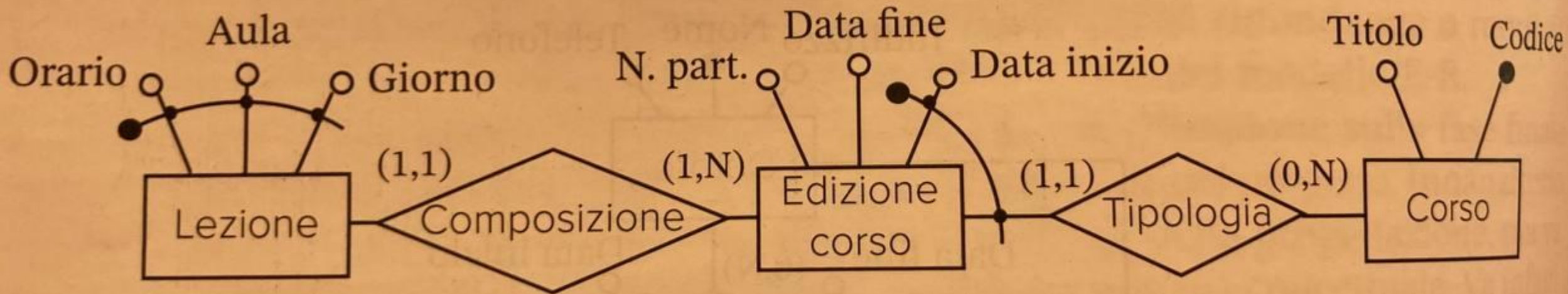


## **Frasi relative ai corsi**

**Per i corsi (circa 200), rappresentiamo il titolo e il codice, le varie edizioni con date di inizio e fine e, per ogni edizione, rappresentiamo il numero di partecipanti e il giorno della settimana, le aule e le ore dove sono tenute le lezioni.**



- Vanno innanzitutto distinti due concetti legati tra loro, ma chiaramente distinti:
  - Il concetto *astratto* di corso (che ha un “nome” e un “codice”)
  - L'*edizione* di un corso (che ha una “data di inizio”, “data di fine”, “num. di partecipanti”)
- Rappresentiamo questi due concetti tramite due entità distinte legate dalla relazione *Tipologia*.
- Poi vanno rappresentate le *Lezioni* dei corsi, che descriviamo come entità legata a *Edizione Corso* tramite la relazione *Composizione*.
- Aggiungiamo poi i vari attributi e i vincoli di partecipazione.
- Scegliamo gli **identificatori**:
  - Assumiamo che una lezione sia identificata da aula, ora, giorno
  - Assumiamo che edizioni diverse dello stesso corso non possano partire nello stesso giorno



# Passo di integrazione e schema finale

Lo *schema finale* si ottiene per *integrazione* degli schemi ottenuti fino a questo punto.

Iniziamo dall'integrazione tra gli schemi relativi ai *docenti* e ai *corsi*. Dall'analisi dei requisiti individuiamo **3 tipi** di correlazioni diverse tra docenti e corsi:

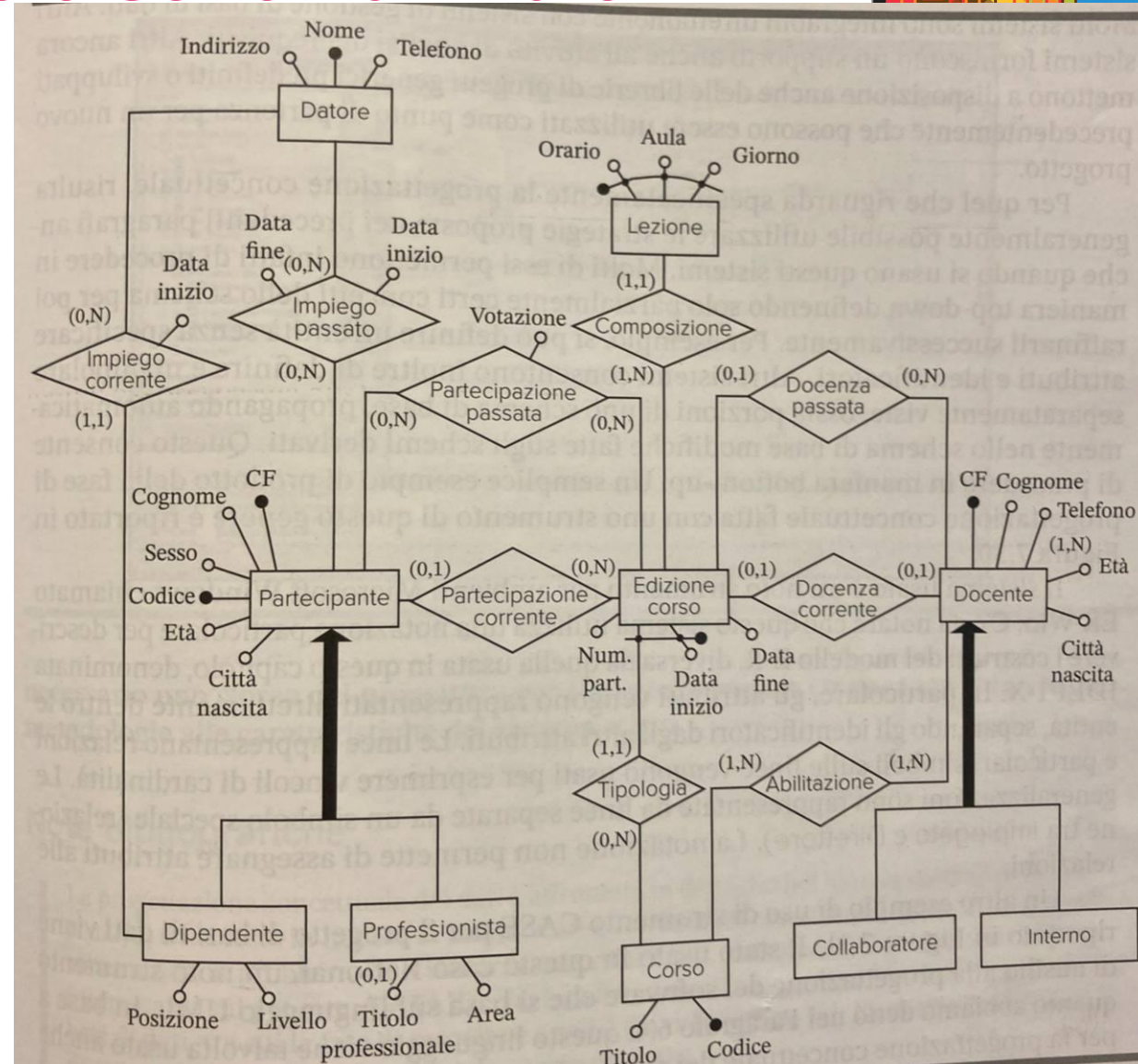
- 1) Docenze correnti
- 2) Docenze passate
- 3) Abilitazione (all'insegnamento di un corso)

Usiamo *tre relazioni*:

- Le prime due collegano Docente a Edizione Corso
- La terza collega Docente a Corso

Aggiungiamo i vincoli di partecipazione.

**Nota:** Docente partecipa con vincolo  $(0, 1)$  alla relazione *Docenza Corrente*, perchè un docente può insegnare (correntemente) al massimo un corso.





## Passo di integrazione e schema finale

Procediamo ora all'integrazione tra *corsi* e *partecipanti*.

Se ne possono individuare due:

- 1) Le partecipazioni *correnti*
- 2) Le partecipazioni *passate*

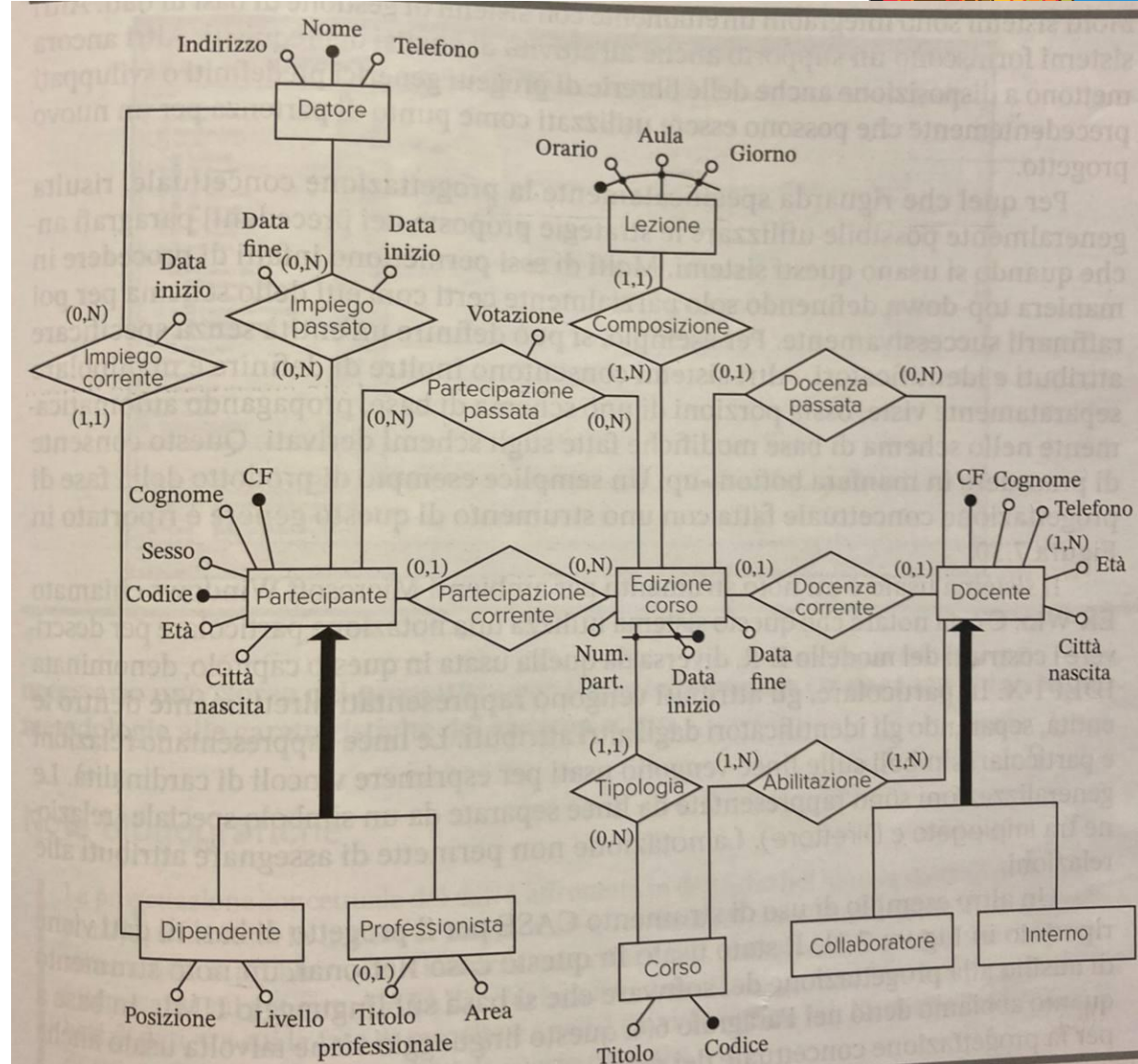
che rappresentiamo con due relazioni tra *Partecipante* e *Edizione Corso*.

Delle partecipazioni passate, è di interesse la *votazione*, che rappresentiamo come un attributo.

Aggiungiamo ora i vari vincoli di partecipazione.

**Nota:** c'è un errore nell'immagine del libro:

- L'entità *Partecipante* partecipa alla relazione *Partecipazione corrente* con vincolo (0,N).
- Vedi frase nelle specifiche: *“le edizioni dei corsi che stanno attualmente frequentando”*.



## Passi finali

A questo punto, restano da verificare le proprietà dello schema così ottenuto.

- Completezza: la verifichiamo ripercorrendo tutti i requisiti sui dati e sulle operazioni
  - Esempio: consideriamo l'**operazione 7**, “*per ogni docente, trova i partecipanti a tutti i corsi da lui/lei insegnati (5 volte alla settimana)*”.
  - Questa operazione è eseguibile sullo schema finale: partiamo dall'entità *Docente*, attraversiamo le relazioni *Docenza Passata* e *Docenza corrente*, raggiungiamo l'entità *Edizione Corso*, e da questa, tramite le relazioni *Partecipazione passata* e *Partecipazione corrente*, arriviamo infine all'entità *Partecipante*.
- Minimalità: esiste nello schema una *ridondanza*, cioè l'attributo “*Numero di partecipanti*” dell'entità *Edizione corso*, che può essere derivato per una certa edizione contando il numero di istanze dell'entità *Partecipante* che sono legate a questa edizione tramite la relazione *Partecipazione corrente*.

**Importante:** è importante descrivere sotto forma di *regole aziendali (vincoli di integrità)* eventuali vincoli non espressi (o non esprimibili) direttamente nello schema.

*Esempio: il fatto che un docente può insegnare un corso solo se è abilitato a farlo.* (non esprimibile in E-R).



# Note

Per la progettazione concettuale, in particolare per disegnare lo schema E-R, potete usare che tool volete.

Dei semplici tool di grafica vettoriale vanno benissimo.

