

# BASI DI DATI I

- Ristrutturazione Class Diagram
- Tavole di analisi
- Passaggi nella ristrutturazione
- Analisi delle ridondanze
- Eliminazione delle generalizzazioni
- Eliminazione attributi multipli
- Eliminazione attributi strutturati
- Partizione/Associazione entità/associazioni
- Identificazione chiavi primarie

# RISTRUTTURAZIONE CLASS DIAGRAM



# OBIETTIVO

- Uno schema logico in grado di descrivere tutte le informazioni contenute nello schema UML prodotto nella fase di progettazione concettuale.



# LA RISTRUTTURAZIONE

- Prima di passare allo schema logico è necessario ristrutturare il class diagram per:
  - Semplificare la traduzione:  
*non tutti i costrutti che abbiamo utilizzato hanno una traduzione naturale nei modelli logici.*
  - Ottimizzare il progetto:  
*rendere più efficiente l'esecuzione delle operazioni.*

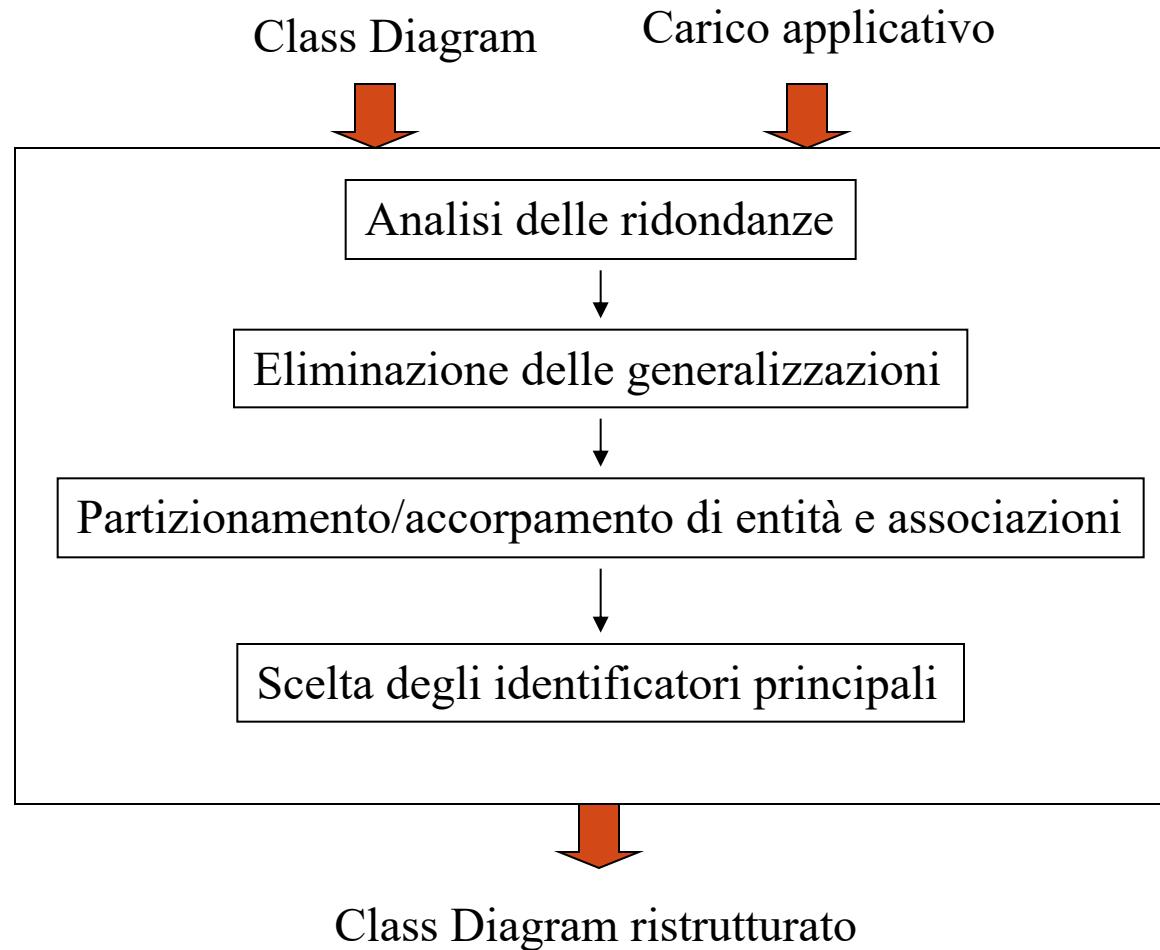


# ATTIVITÀ DI RIORGANIZZAZIONE E TRADUZIONE

- Ristrutturazione del Class Diagram UML:
  - *si basa su criteri di ottimizzazione dello schema.*
- Traduzione verso il modello logico:
  - *fa riferimento ad uno specifico modello logico.*
- I dati in ingresso sono lo schema concettuale ed il carico applicativo (dimensione dei dati e caratteristiche delle operazioni).



# RISTRUTTURAZIONE DI SCHEMI LOGICI



# ANALISI DELLE PRESTAZIONI

- Indici di prestazioni (non valutabili in maniera precisa in sede di progettazione logica):
  - Costo di una operazione (# occorrenze di entità e relazioni visitate per rispondere a una operazione).
  - Occupazione di memoria.

Per studiare questi parametri è necessario conoscere:



- Volume dei dati:
  - # occorrenze di entità e relazioni.
  - Dimensioni di ciascun attributo.
- Caratteristiche delle operazioni:
  - Tipo (interattiva o batch).
  - Frequenza (# medio di esecuzioni in un  $t$ ).
  - Dati coinvolti.



# TAVOLE DI ANALISI



# TAVOLE DI ANALISI

- Tavola dei volumi:

## **concetto tipo volume**

- Vengono riportati tutti i concetti dello schema (entità e relazioni) con il volume previsto a regime.

- Tavola delle operazioni:

## **operazioni tipo frequenza**

- Viene riportata per ogni operazione la frequenza prevista, il tipo (interattiva o batch).

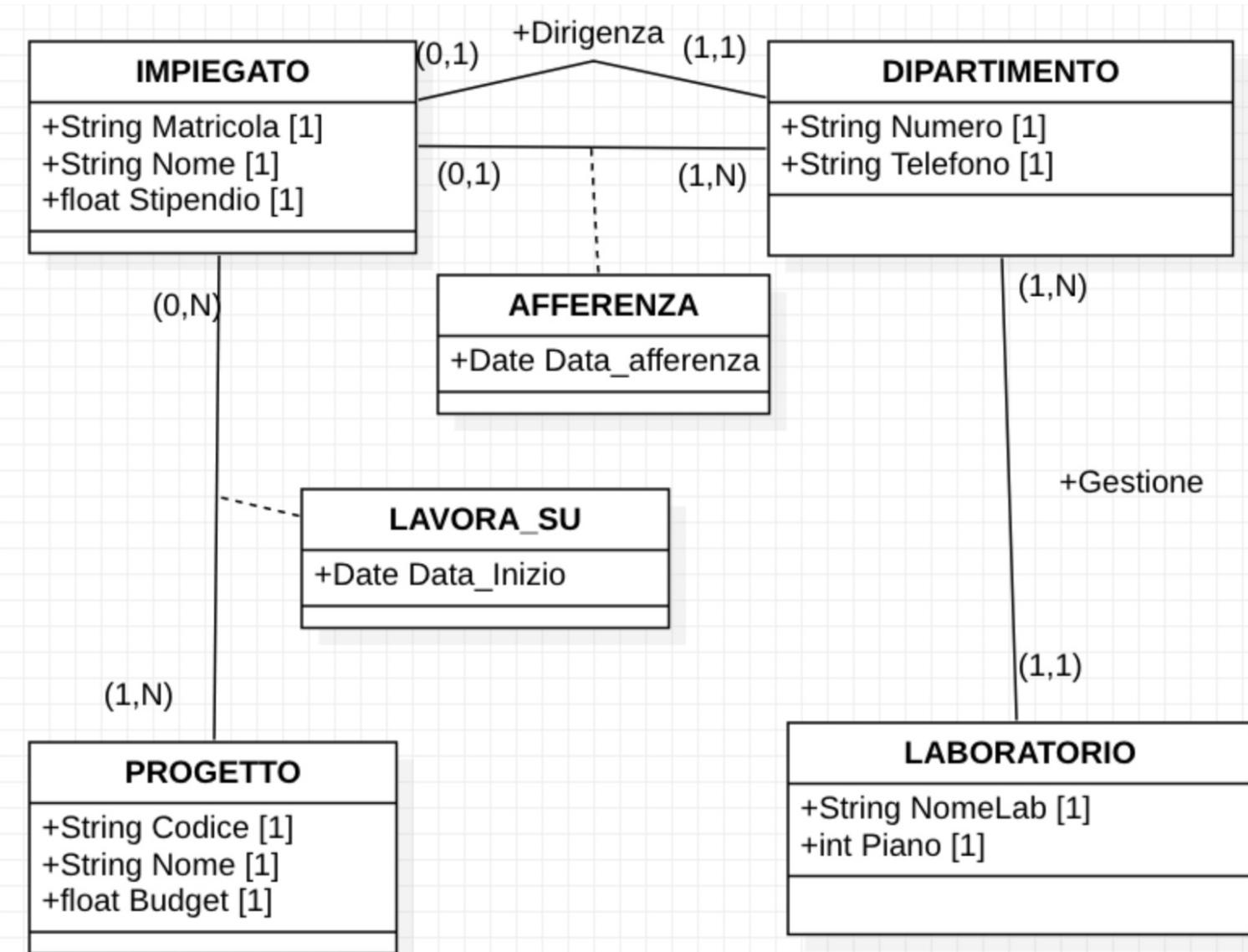
- Tavola degli accessi

## **concetto costrutto accessi tipo**

- Viene riportata, per ogni operazione, il numero di accessi ai concetti coinvolti ed il tipo di accesso (L/S).
- Le operazioni in scrittura (S) sono più onerose di quelle in lettura (L).



# ESEMPIO



# TAVOLA DEI VOLUMI

Concetto	Tipo	Volume
Laboratorio	E	100
Dipartimento	E	80
Impiegato	E	2000
Progetto	E	500
Gestione	R	100
Afferenza	R	1900
Dirigenza	R	80
Lavora_Su	R	6000



# **NUMERO DELLE OCCORRENZE DELLE ASSOCIAZIONI**

- Nella tavola dei volumi.
- Dipende da due parametri:
  - Numero delle occorrenze delle entità coinvolte nelle associazioni.
  - Numero medio di partecipazioni di un'occorrenza di entità alle occorrenze di associazioni.  
**Es:** supponiamo che un impiegato lavori in media a tre progetti.



# OPERAZIONI

1. Assegna un impiegato ad un progetto.
2. Trova i dati di un impiegato, del dipartimento nel quale lavora e dei progetti ai quali partecipa.
3. Trova i dati di tutti gli impiegati di un certo dipartimento.
4. Per ogni sede, trova i suoi dipartimenti con il cognome del direttore e l'elenco degli impiegati del dipartimento.



# TAVOLA DELLE OPERAZIONI

Operazione	Tipo	Frequenza
Op. 1	I	50 al giorno
Op. 2	I	100 al giorno
Op. 3	I	10 al giorno
Op. 4	B	2 a settimana

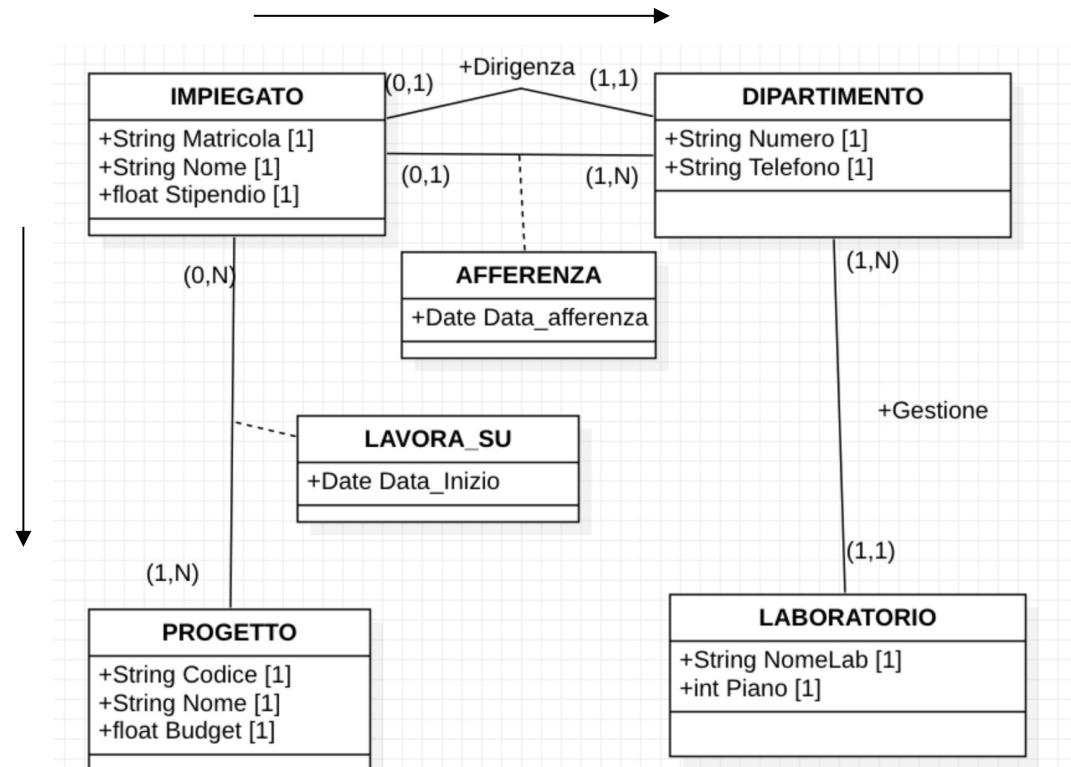


# SCHEMA DI OPERAZIONE

- Per ogni operazione possiamo descrivere graficamente i dati coinvolti attraverso uno **schema di operazione**.
- Descrive il cammino logico da percorrere per accedere alle informazioni di interesse.

## Operazione 2

Trova i dati di un impiegato, del dipartimento nel quale lavora e dei progetti ai quali partecipa.



# COSTO DI UN'OPERAZIONE

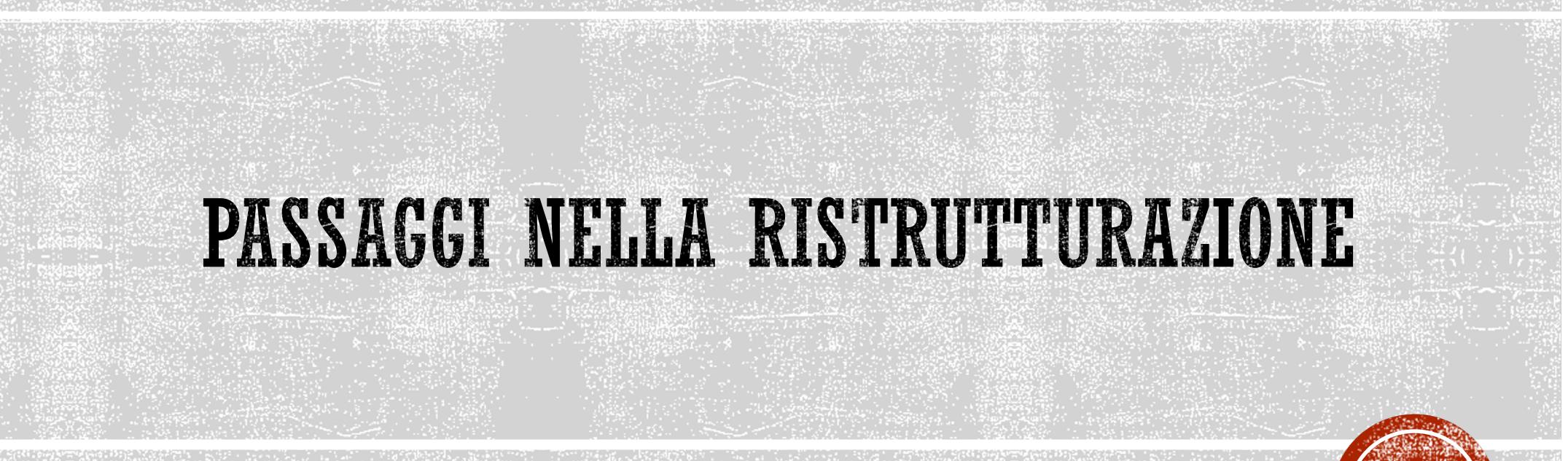
- Si stima il costo di un'operazione contando il numero degli accessi alle occorrenze di entità e di relazioni.



# TAVOLA DEGLI ACCESSI (RIF. OP. 2)

Concetto	Costrutto	Accessi	Tipo
Impiegato	E	1	L
Afferisce	R	1	L
Dipartimento	E	1	L
Partecipa	R	3	L
Progetto	E	3	L





# PASSAGGI NELLA RISTRUTTURAZIONE



# RISTRUTTURAZIONE DEL CLASS DIAGRAM

- La fase di ristrutturazione di uno schema concettuale è suddiviso:
  1. Analisi delle ridondanze.
  2. Eliminazione delle generalizzazioni.
  3. Eliminazione Attributi Multivalore
  4. Eliminazione Attributi Strutturati
  5. Partizionamento/accorpamento di entità e associazioni.
  6. Scelta degli identificatori primari.



# **ANALISI DELLE RIDONDANZE**



# ANALISI DELLE RIDONDANZE

- Una **ridondanza** in uno schema concettuale corrisponde alla presenza di un dato che può essere derivato da altri dati.
- **Vantaggi**
  - Semplificazione delle interrogazioni.
- **Svantaggi**
  - Appesantimento degli aggiornamenti.
  - Maggiore occupazione di spazio.

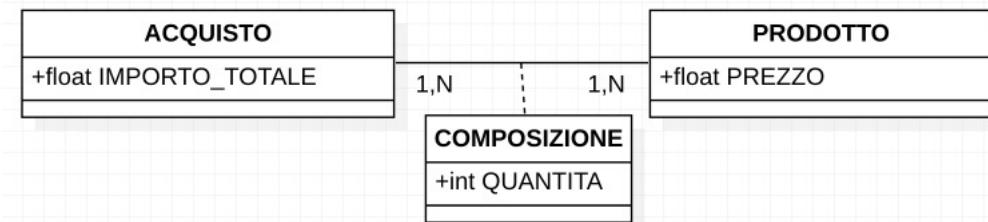


# ATTRIBUTI DERIVABILI

- Da attributi della stessa entità /associazione:



- Da attributi di altre entità/associazioni

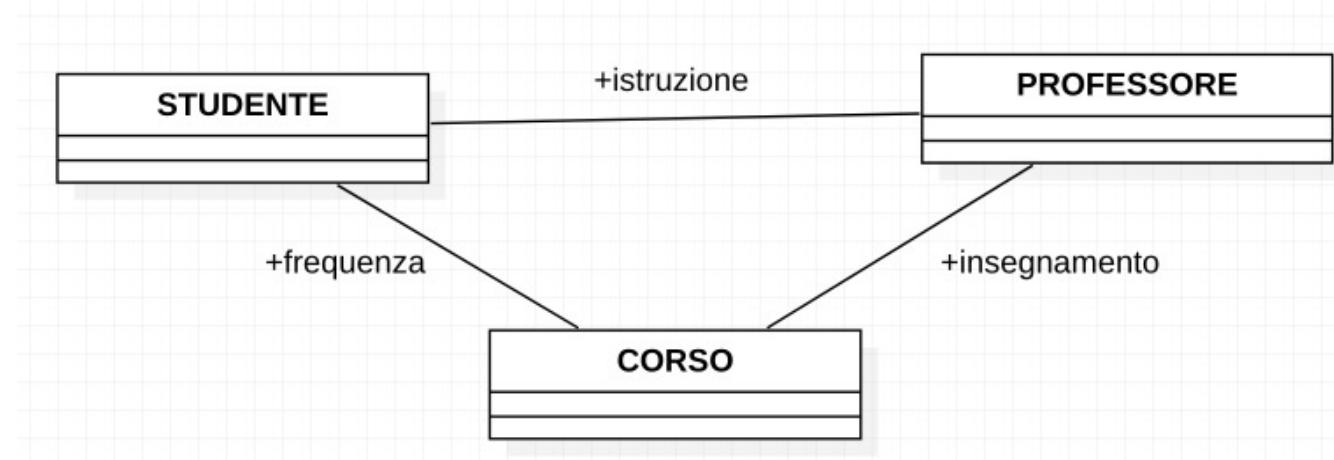


- Dal conteggio di occorrenze



# ATTRIBUTI DERIVABILI (4)

- Da associazioni in presenza di cicli:



- La presenza di cicli non genera necessariamente ridondanze:
  - *Si immagini che l'associazione "istruzione" fosse "è tesista di".*



# INSERIRE UNA RIDONDANZA?

- La decisione va presa confrontando il costo delle operazioni che coinvolgono il dato ridondante e relativa occupazione di memoria nei casi di **presenza/assenza** di ridondanza.



# TAVOLE DI ANALISTI

- Tavola dei volumi:

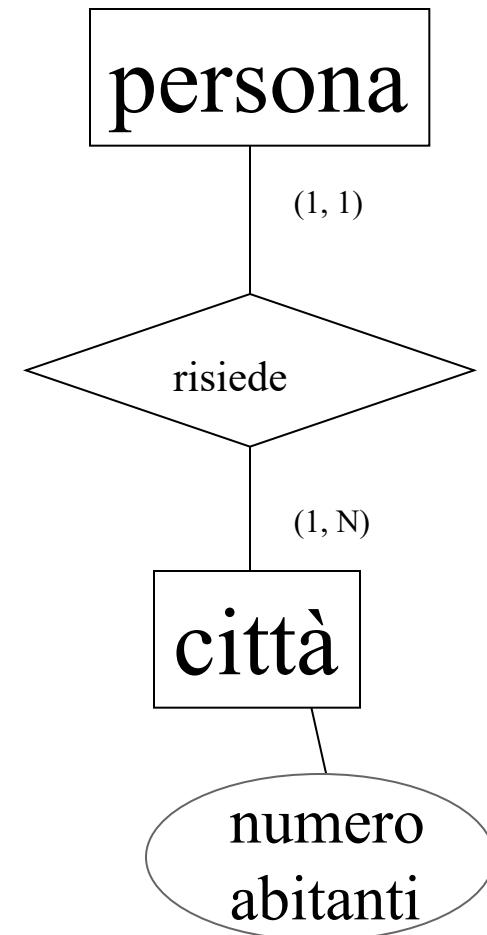
▪ Città	E	200
▪ Persona	E	1000000
▪ Risiede	R	1000000

- *Op<sub>1</sub>: memorizza una nuova persona con la relativa città di residenza.*

- *Op<sub>2</sub>: stampa tutti i dati di una città (incluso il numero di abitanti).*

- Tavola delle operazioni:

▪ Op <sub>1</sub>	I	500/giorno
▪ Op <sub>2</sub>	I	2/giorno



# **“NUMERO DI ABITANTI” → CITTÀ**

- Proviamo a valutare gli indici di prestazione in caso di presenza del dato ridondante:
- $\text{Mem}(\text{num\_abit}) \approx 4 \text{ byte} \rightarrow \text{dato ridondante} \approx 4 \times 200 = 800 \text{ byte} (\approx 1 \text{ kbyte})$
- Costo:*Un accesso in scrittura ha un costo doppio rispetto ad un accesso in lettura*
  - .



# TAVOLE DEGLI ACCESSI (CON RIDONDANZA)

- Operazione 1: *memorizza una nuova persona con la relativa città di residenza.*

- Persona      E      1      S

(per memorizzare una nuova persona)

- Risiede      R      1      S

(per memorizzare una nuova coppia città-persona)

- Città      E      1      L

(per cercare la città di interesse)

- Città      E      1      S

(incrementa di 1 il numero di abitanti)

- $3S * 500/\text{giorno} + 1L * 500/\text{giorno}$

↓

- 3500 accessi/giorno

(un accesso in scrittura vale il doppio  
di un accesso in lettura)

- Operazione 2: *stampa tutti i dati di una città (incluso il numero di abitanti).*

- Città      E      1      L

- Trascurabile:  $1L * 2/\text{giorno}$



# TAVOLE DEGLI ACCESSI (SENZA RIDONDANZA)

- Op.1: memorizza una nuova persona con la relativa città di residenza.

- Persona E l S
- Risiede R l S

(non si accede a Città per aggiornare il dato derivato)

- $2S * 500/\text{giorno}$



- 2000 accessi/giorno

(contano doppio gli accessi in scrittura)

- Op.2: stampa tutti i dati di una città (incluso il numero di abitanti).

- Città E l L
- Risiede R 5000<sup>1</sup> L

- $1L + 1L * 5000 * 2/\text{giorno}$



- 10000 accessi/giorno

<sup>1</sup> (#abitanti/città=1000000/200)

- 
- **12.000 accessi vs 3.500 accessi + 1Kbyte**
  - Gli accessi in lettura necessari per calcolare il dato derivato sono molti di più degli accessi in scrittura per mantenerlo aggiornato.



# **ELIMINAZIONE DELLE GENERALIZZAZIONI**



# ELIMINAZIONE DELLE GENERALIZZAZIONI

- I sistemi tradizionali per la gestione di basi di dati non consentono di rappresentare direttamente una generalizzazione.
- È necessario tradurre le generalizzazioni usando altri costrutti dell'UML(classi, associazioni).

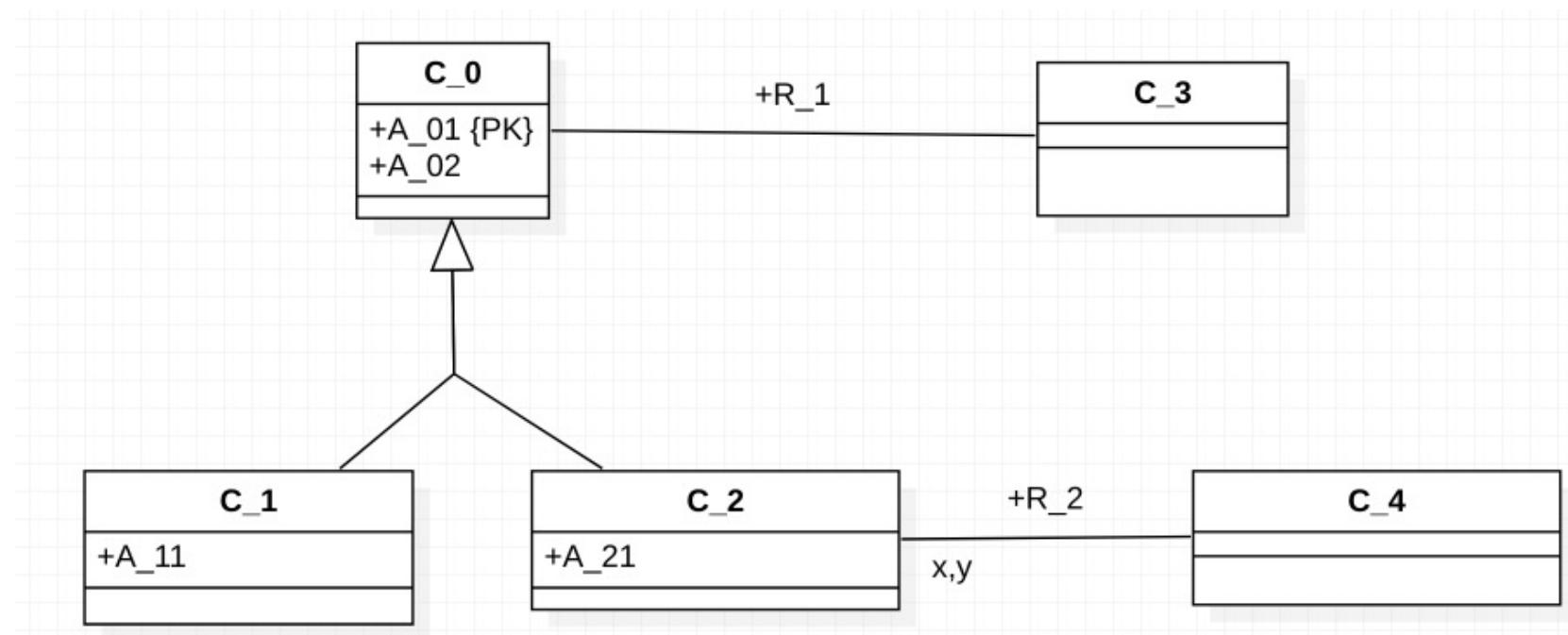


# ELIMINAZIONE DELLE GENERALIZZAZIONI

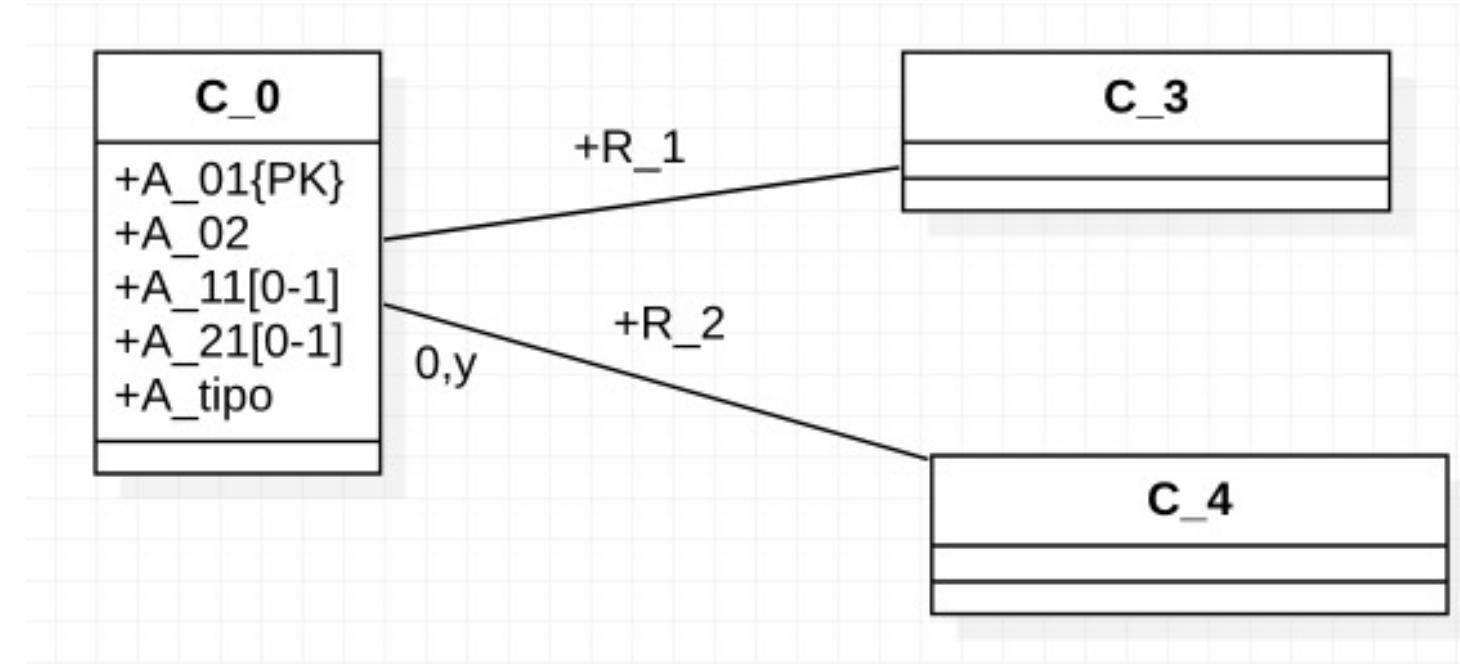
- a) Accorpamento delle figlie della generalizzazione nel padre.
- b) Accorpamento del padre della generalizzazione nelle figlie.
- c) Sostituzione della generalizzazione con associazioni.



# SCHEMA INIZIALE



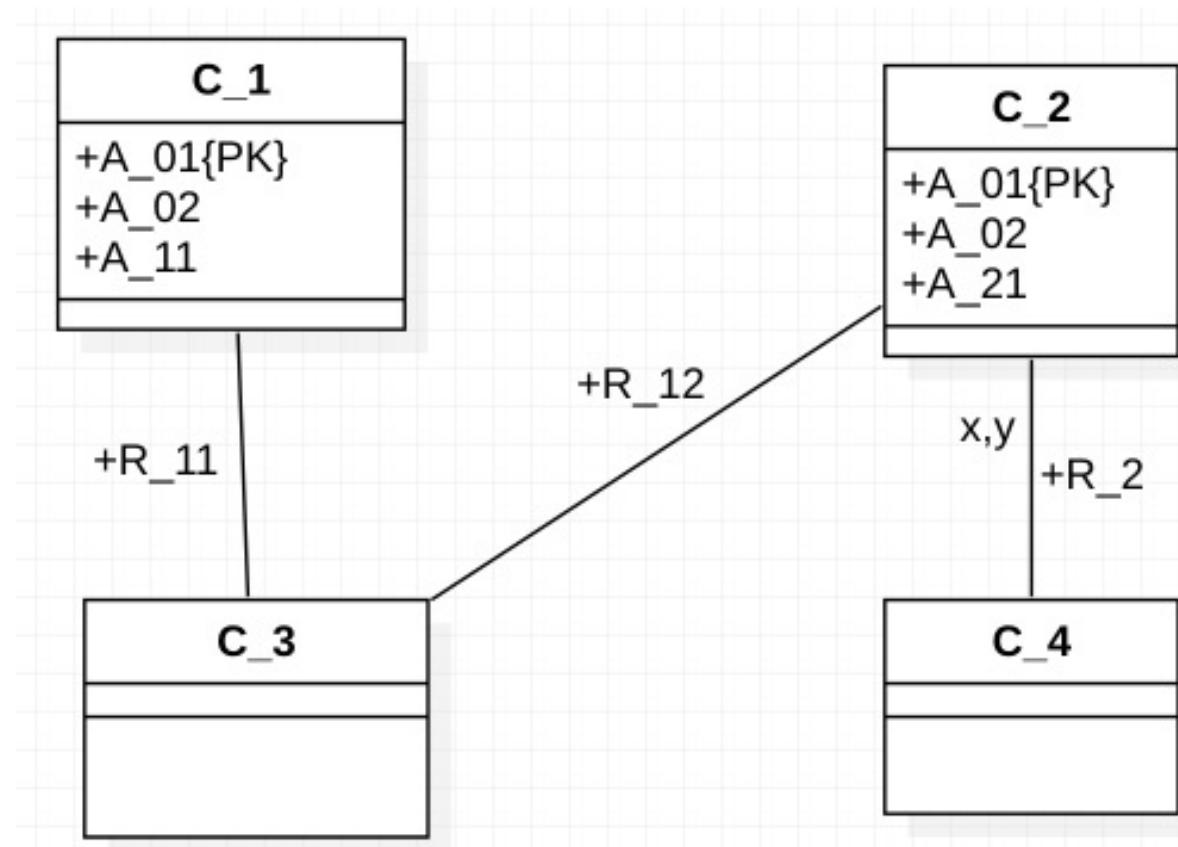
## (A) ACCORPAMENTO DELLE FIGLIE DELLA GENERALIZZAZIONE NEL PADRE



Le entità C<sub>1</sub> e C<sub>2</sub> vengono eliminate e le loro proprietà vengono aggiunte al padre C<sub>0</sub>.

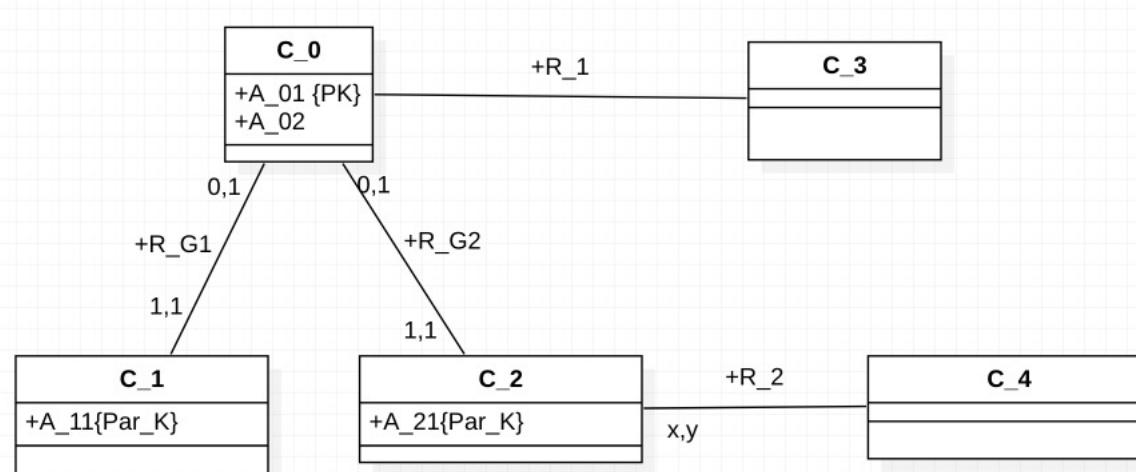


## (B) ACCORPAMENTO DEL PADRE DELLA GENERALIZZAZIONE NELLE FIGLIE



## (C) SOSTITUZIONE DELLA GENERALIZZAZIONE CON ASSOCIAZIONI

- Le entità  $C_1$  e  $C_2$  sono identificate esternamente dall'entità  $C_0$ :
- **Vincoli:** ogni occorrenza di  $C_0$  non può partecipare contemporaneamente a  $R_{G1}$  ed  $R_{G2}$  se la generalizzazione è totale, ogni occorrenza di  $E_0$  deve partecipare o ad un'occorrenza di  $R_{G1}$  o ad un'occorrenza di  $R_{G2}$



- **Vincolo di totalità:** Un elemento di  $C_0$  è associato ad un elemento di  $C_1$  o di  $C_2$ 
  - (**Parziale** se un elemento di  $C_0$  non è associato a nessuno dei figli)
- **Vincolo di disgiunzione:** Un elemento di  $C_0$  non può essere associato a entrambi i figli
  - (**Overlapping** se un elemento può essere associato a entrambi i figli)



# COME SCEGLIERE TRA LE DIVERSE ALTERNATIVE

- L'alternativa **(a)** conviene quando le operazioni non fanno molta distinzione tra le occorrenze e tra gli attributi di  $C_0$ ,  $C_1$ , ed  $C_2$ . Introduciamo valori nulli, ma abbiamo un minor numero di accessi.
- L'alternativa **(b)** è applicabile quando la generalizzazione è totale. Altrimenti le occorrenze di  $C_0$  che non sono occorrenze né di  $C_1$ , né di  $C_2$  non sarebbero rappresentate.
- L'alternativa **(c)** è applicabile quando la generalizzazione non è totale, e ci sono operazioni che fanno distinzione tra entità padre ed entità figlie. Assenza di valori nulli e incremento degli accessi.



## COME SCEGLIERE TRA LE DIVERSE ALTERNATIVE (2)

- La ristrutturazione delle generalizzazioni è un tipico caso per cui il conteggio delle istanze e degli accessi non permette sempre di scegliere la migliore alternativa.
- Infatti, sembrerebbe che l'alternativa **(c)** non conviene quasi mai perché richiede molti più accessi per eseguire le operazioni sui dati.
- Questo tipo di ristrutturazione ha il grosso vantaggio di generare entità con pochi attributi:
  - A livello pratico questo si traduce in strutture logiche di piccole dimensioni per cui un accesso fisico permette di recuperare molti dati (tuple) in una sola volta.



# **GENERALIZZAZIONI A PIÙ LIVELLI**

- Per le generalizzazioni a più livelli si può procedere analizzando una generalizzazione alla volta a partire dal fondo della gerarchia.

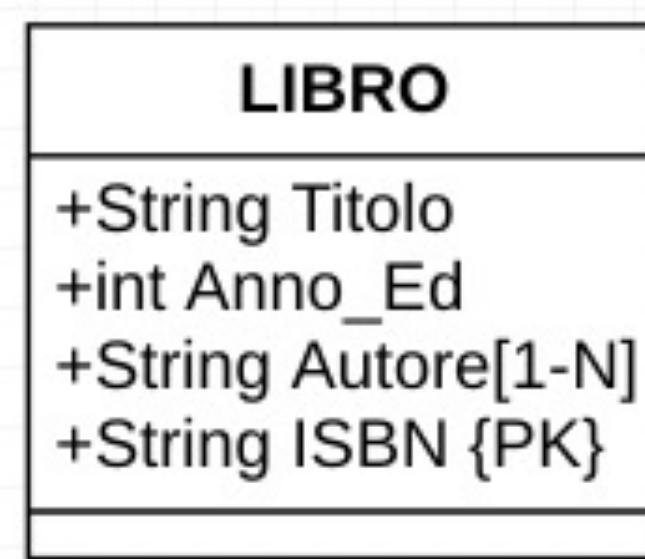


# **ELIMINAZIONE ATTRIBUTI MULTIVALORE**

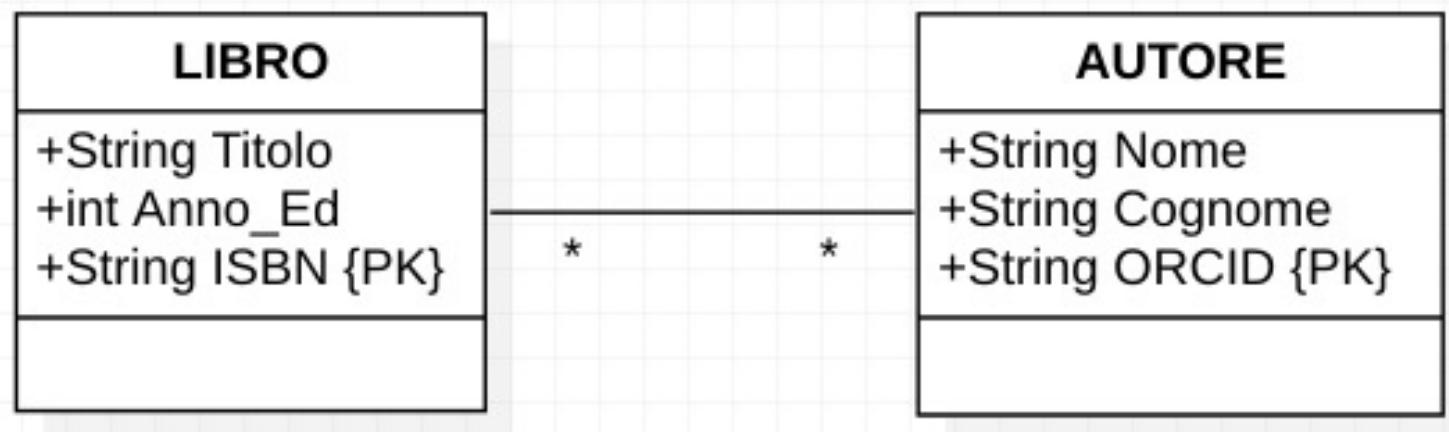


# ELIMINAZIONE ATTRIBUTI MULTIVALORE

- Il modello relazionale non consente questa rappresentazione:



# 1) CREAZIONE DI UNA ENTITÀ ESTERNA ASSOCIATA



## 2) TRATTARE L'ATTRIBUTO MULTIPLO COME FOSSE SINGOLO



Titolo: «Sistemi di basi di dati»  
Anno\_Ed: «2015»  
ISBN: «6767-8754»  
Autori: «Elmasri, Navathe»



### 3) REPLICARE L'ATTRIBUTO NELLA CLASSE (QUANTE VOLTE??)



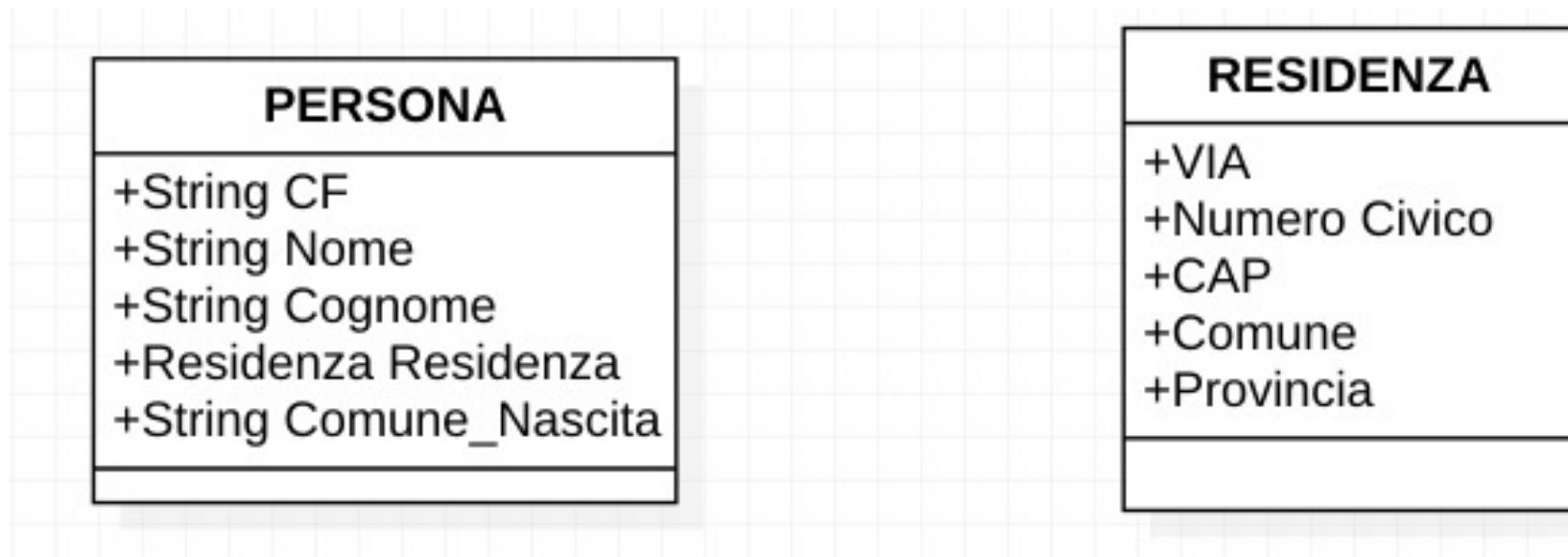
- Per alcuni casi potrebbe anche andare bene.
- Persona-> Telefono1, Telefono2
- Per un numero limitato di volte potrebbe essere una soluzione accettabile
  - Andare troppo oltre potrebbe causare un numero eccessivo di campi vuoti



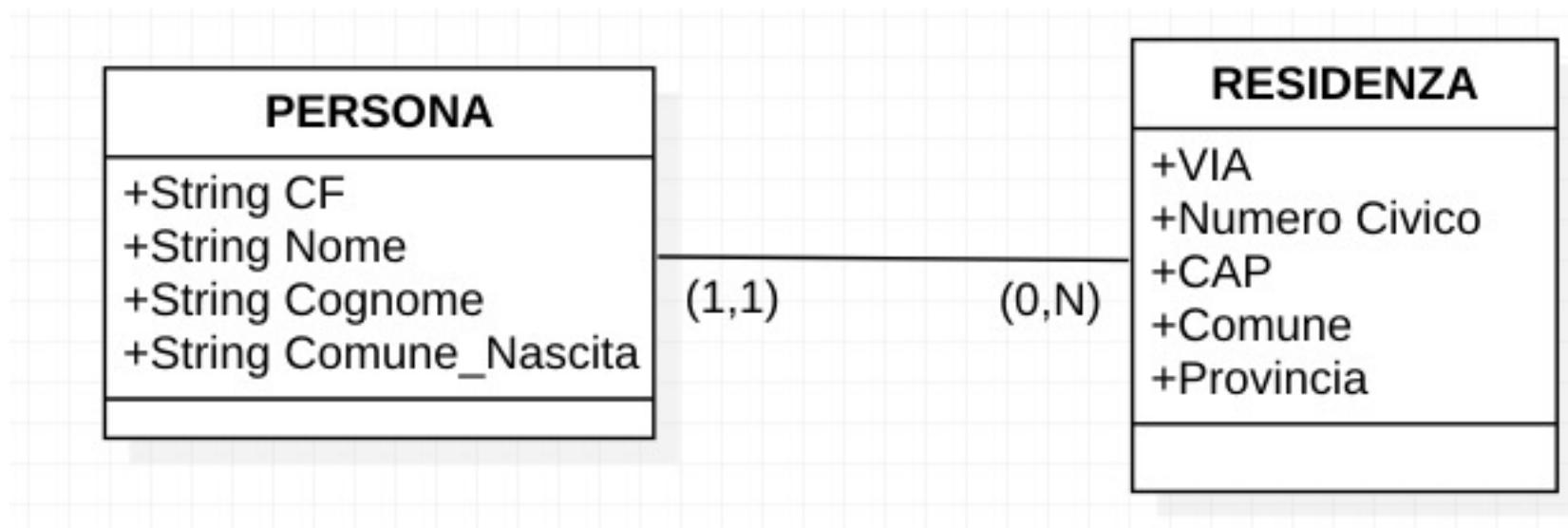
# **ELIMINAZIONE ATTRIBUTI STRUTTURATI**



# ELIMINAZIONE ATTRIBUTI STRUTTURATI



# 1) INTRODUZIONE DI UNA CLASSE PER L'ATTRIBUTO STRUTTURATO



## 2) ESTRAZIONE DEGLI ATTRIBUTI DELLA CLASSE

PERSONA
+String CF
+String Nome
+String Cognome
+String Comune_Nascita
+String Via
+String Numero_Civico
+String CAP
+String Comune
+String Provincia



# 3) TRASCURARE LA STRUTTURA DELL'ATTRIBUTO

<b>PERSONA</b>
+String CF
+String Nome
+String Cognome
+String Comune_Nascita
+String Residenza



# **PARTIZIONAMENTO/ACCORPAMENTO DI ENTITÀ/ASSOCIAZIONI**

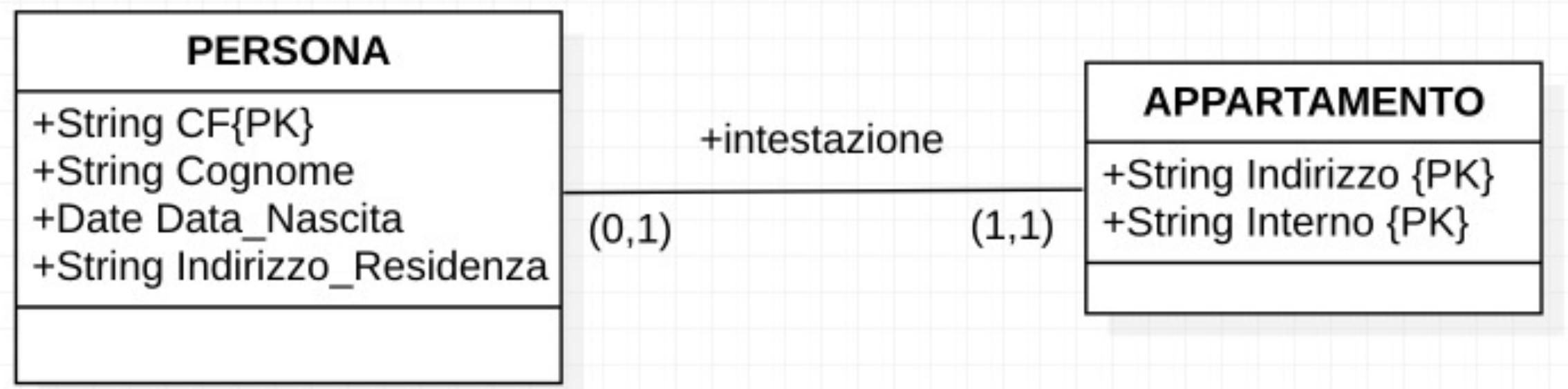


# ACCORPAMENTI DI ENTITÀ

- Accorpamento dei concetti operando sulla struttura.
- È conveniente quando le operazioni accedono a dati presenti su entrambe le entità.
- Si effettua generalmente su associazioni di tipo 1:1.
- Presenza di valori nulli.



# ACCORPAMENTI DI ENTITÀ - ESEMPIO



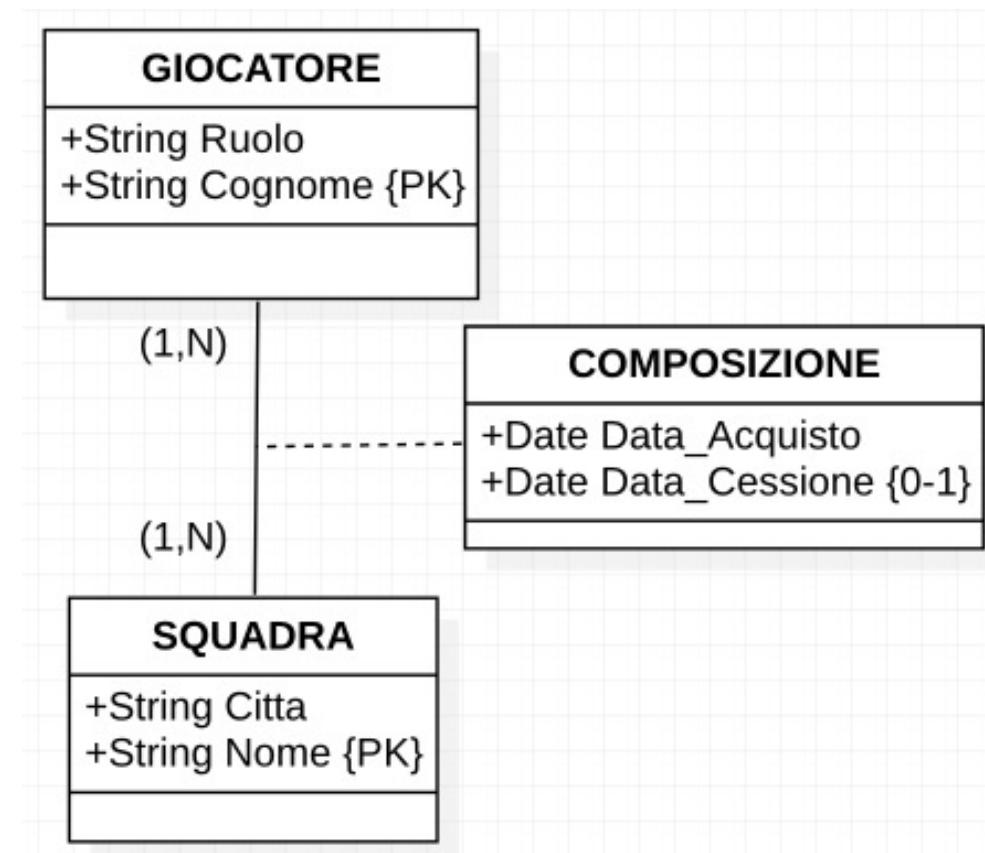
# ACCORPAMENTI DI ENTITÀ – ESEMPIO (2)

- Le operazioni più frequenti su persona richiedono sempre dati relativi all'appartamento che occupa:

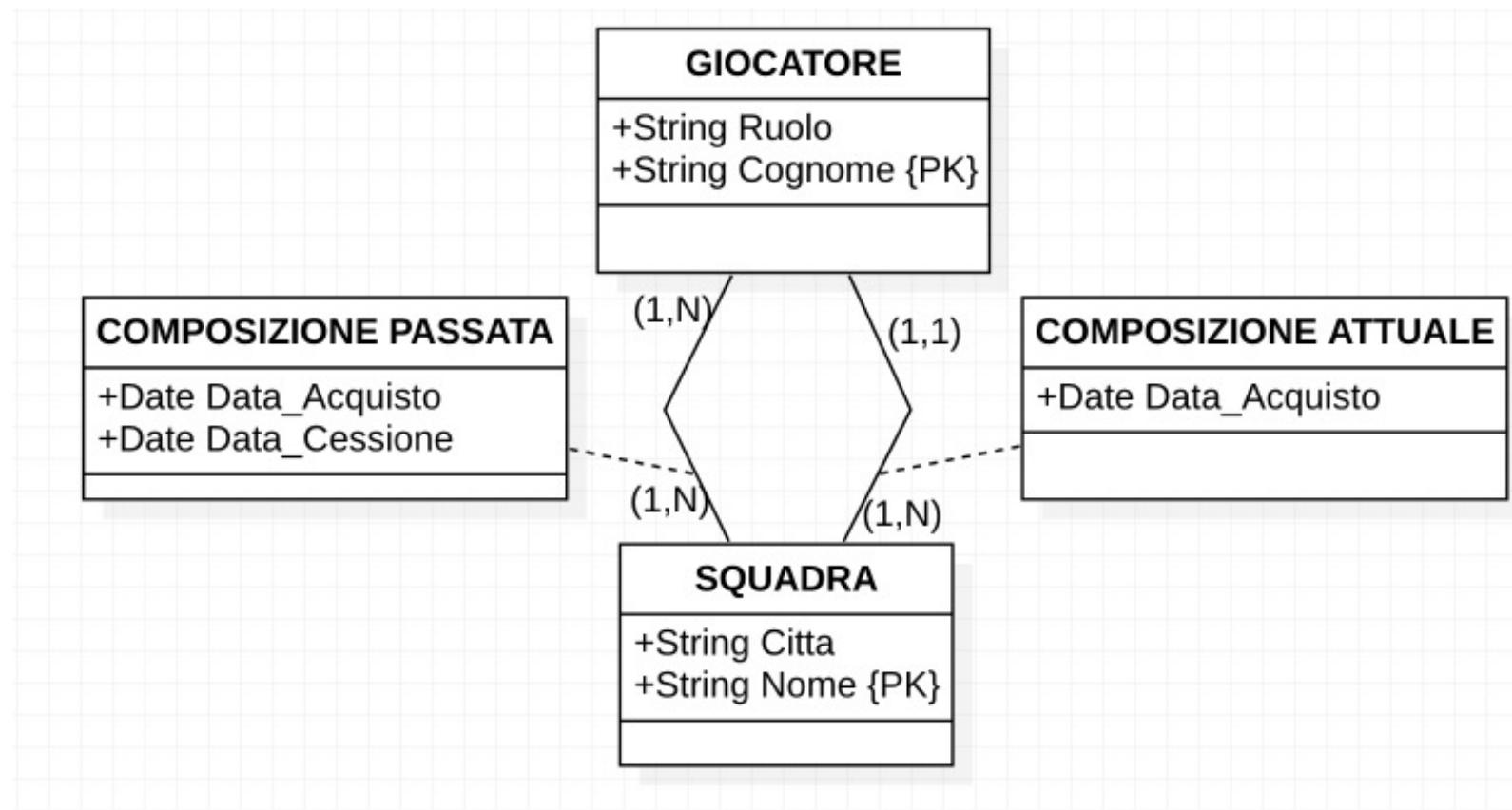
PERSONA
+String CF{PK}
+String Cognome
+String Data_Nascita
+String Indirizzo_Residenza
+String Indirizzo_Appartamento{0-1}
+String Interno_Appartamento{0-1}



# DECOMPOSIZIONE DI ASSOCIAZIONI - ESEMPIO



# DECOMPOSIZIONE DI ASSOCIAZIONI – ESEMPIO (2)



# **IDENTIFICATORE CHIAVI PRIMARIE**



# IDENTIFICATORI

- Scelta della chiave primaria per la costruzione degli indici per il recupero efficiente dei dati.
- Criteri generali:
  - Escludere attributi con valori nulli.
  - Numero minimo di attributi (dimensioni ridotte degli indici).
  - Identificatore coinvolto in molte operazioni.
  - Velocità di accesso all'indice.
    - Tempo minimo per confrontare i valori.
- Se nessuno degli identificatori candidati soddisfa tali criteri si introduce un ulteriore attributo all'entità:
  - Questo attributo conterrà valori speciali (**codici**) generati appositamente per identificare le occorrenze delle entità.



# IDENTIFICATORI: ESEMPIO

- Nel caso in cui l'entità Studente abbia **due** possibili chiavi:
  1. Matricola (10 caratteri, numerici).
  2. Codice fiscale (16 caratteri, alfanumerici).
- Perché è opportuno selezionare l'attributo *Matricola* come identificatore primario?
  - *La matricola richiede un tempo minore per confrontare due valori tra loro: la velocità di accesso all'indice è ridotta*





# FINE

Per eventuali domande: (in ordine di preferenza personale)

- Ora.
- Chat di Teams
- Mail: [silvio.barra@unina.it](mailto:silvio.barra@unina.it)

