

Basi di dati

Giuseppe De Giacomo

*Dipartimento di Informatica e Sistemistica "Antonio Ruberti"
Università di Roma "La Sapienza"*

Anno Accademico 2005/2006
Canale M-Z

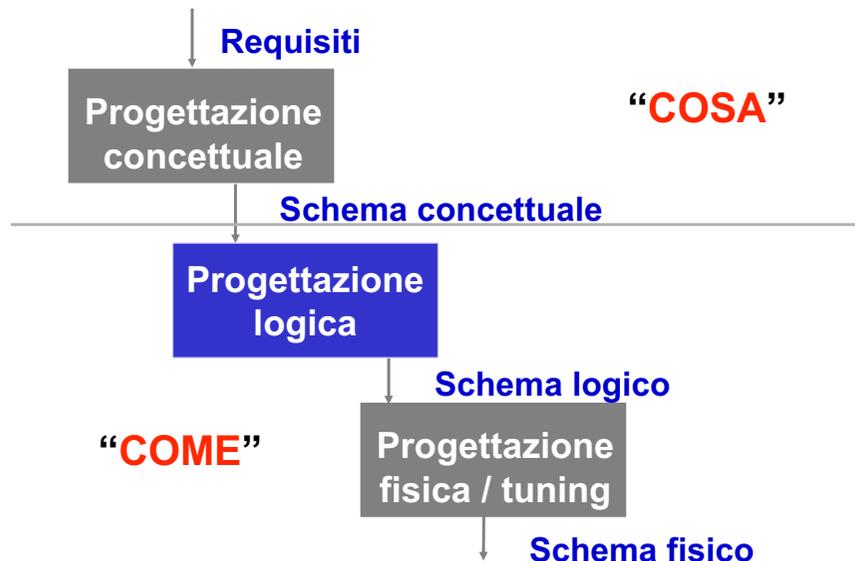
<http://www.dis.uniroma1.it/~degiacomo/didattica/basidati/>

5. La progettazione logica

5.1 introduzione alla progettazione logica

1. **introduzione alla progettazione logica**
2. ristrutturazione dello schema ER
3. traduzione diretta nel modello relazionale
4. ristrutturazione dello schema logico

Fasi della progettazione



Obiettivo della progettazione logica

Tradurre lo schema concettuale (espresso nel modello ER con vincoli) in uno schema logico che rappresenti gli stessi dati:

- utilizzando il **modello logico** del DBMS scelto (nel nostro caso, il modello relazionale con vincoli)
- in maniera **corretta, completa ed efficiente**

Dati di ingresso e uscita

Ingresso:

- schema concettuale (diagramma ER e dizionario dei dati)
- informazioni sul carico applicativo

Uscita:

- schema logico (nel modello relazionale)
- vincoli aggiuntivi
- documentazione associata

Non si tratta di una pura e semplice traduzione

Motivi:

- Alcuni aspetti dello schema ER possono non essere direttamente rappresentabili nel modello relazionale. È quindi opportuno ristrutturare lo schema ER in modo da renderlo traducibile in modo diretto.
- È necessario porre attenzione alle **prestazioni**.
- Adottiamo un semplice **modello di costo** che permette di fornire una valutazione approssimata delle prestazioni della base di dati in funzione di un certo carico applicativo.
- Le scelte durante in fase di progettazione logica devono essere effettuate con l'obiettivo di ottimizzare le prestazioni.

Carico applicativo

Consideriamo degli "indicatori" dei parametri che regolano le prestazioni:

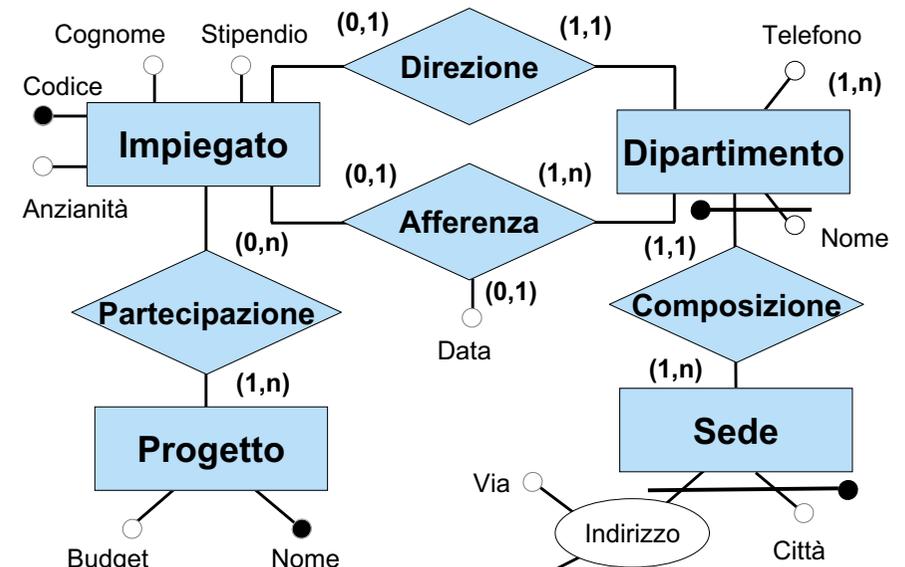
- **tempo di esecuzione** delle operazioni di principale interesse: numero di istanze (di entità e relazioni) mediamente accedute durante l'esecuzione dell'operazione (**accessi**)
- **spazio di memoria** necessario per memorizzare i dati di interesse

Per valutare questi parametri bisogna conoscere (oltre allo schema):

- **volume dei dati:**
 - numero di istanze previste di entità e relazioni
 - dimensione di ciascun attributo
- **caratteristiche delle operazioni:**
 - tipo: interattiva o batch
 - frequenza: numero medio di esecuzioni in un certo periodo
 - dati coinvolti

Si noti che la valutazione sarà necessariamente approssimata, in quanto le prestazioni effettive della base di dati dipendono anche da parametri fisici, difficilmente prevedibili in questa fase (DBMS utilizzato, indici, ...).

Esempio di carico applicativo: schema



Esempio di carico applicativo: operazioni

Supponiamo che le operazioni di interesse siano:

1. Assegna un impiegato ad un progetto.
2. Trova tutti 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.

Tabella dei volumi e tavola delle operazioni

Tabella dei volumi

| Concetto | Costrutto | Volume |
|----------------|-----------|--------|
| Sede | Entità | 10 |
| Dipartimento | Entità | 80 |
| Impiegato | Entità | 200 |
| Progetto | Entità | 500 |
| Composizione | Relazione | 80 |
| Afferenza | Relazione | 190 |
| Direzione | Relazione | 80 |
| Partecipazione | Relazione | 600 |

Tabella delle operazioni

| Op. | Tipo | Frequenza |
|-----|-------------|--------------|
| 1 | Interattiva | 50 / giorno |
| 2 | Interattiva | 100 / giorno |
| 3 | Interattiva | 10 / giorno |
| 4 | Batch | 2 / sett. |

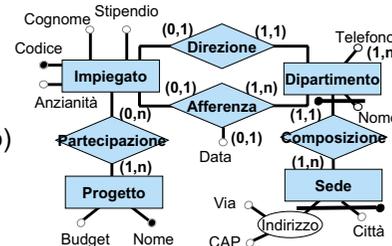
Tabella dei volumi

Si noti che i valori relativi al numero di istanze di entità e relazioni nella tabella dei volumi sono influenzati:

- dalle cardinalità nello schema
- dal numero medio di volte che le istanze delle entità partecipano alle relazioni

Esempio:

- $\text{vol}(\text{Composizione}) = \text{vol}(\text{Dipartimento})$
- $\text{vol}(\text{Direzione}) = \text{vol}(\text{Dipartimento})$
- $\text{vol}(\text{Afferenza}) \leq \text{vol}(\text{Impiegato})$
- se ogni impiegato partecipa in media a 3 progetti:
 $\text{vol}(\text{Partecipazione}) \approx 3 \times \text{vol}(\text{Impiegato})$



Valutazione di costo

Per valutare il costo di un'operazione, si costruisce una **tabella degli accessi** basata su uno schema di navigazione associato all'operazione.

Esempio: trova tutti i dati di un impiegato, del dipartimento nel quale lavora e dei progetti ai quali partecipa (operazione 2).

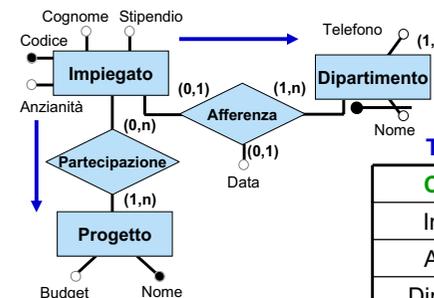


Tabella degli accessi dell'operazione 2

| Concetto | Costrutto | Accessi | Tipo |
|----------------|-----------|---------|------|
| Impiegato | Entità | 1 | L |
| Afferenza | Relazione | 1 | L |
| Dipartimento | Entità | 1 | L |
| Partecipazione | Relazione | 3 | L |
| Progetto | Entità | 3 | L |

Valutazione dei costi e traduzione

- La traduzione dello schema concettuale in uno schema relazionale è guidata dall'**obiettivo di ottimizzare le prestazioni**.
- La valutazione delle prestazioni può essere effettuata adottando il modello di costo appena visto.
- Alcune scelte del processo di traduzione sono di fatto fisse e dettate dalla struttura dello schema ER (e quindi dalle scelte effettuate in fase di progettazione concettuale). In determinati casi invece il progettista deve effettuare delle scelte volte a ottimizzare le prestazioni.
- Vediamo una metodologia di progettazione, articolata in diverse fasi, nella quale i momenti in cui il progettista deve effettuare scelte progettuali sono chiaramente delimitati.

Fasi della progettazione logica

1. Ristrutturazione dello schema ER:

- eliminazione dei costrutti non direttamente traducibili nel modello relazionale
- scelta degli identificatori principali delle entità

2. Traduzione diretta dello schema ER ristrutturato nel modello relazionale:

- la traduzione è **diretta**, nel senso che non richiede (quasi) scelte da parte del progettista
- lo schema relazionale prodotto **non** contiene ridondanze, se non quelle volute
- la traduzione diretta tiene conto delle scelte fatte in fase di progettazione concettuale

3. Ristrutturazione dello schema relazionale:

- richiede delle scelte da parte del progettista, tenendo conto delle prestazioni (carico applicativo)

5. La progettazione logica

5.2 ristrutturazione dello schema ER

1. introduzione alla progettazione logica
- 2. ristrutturazione dello schema ER**
3. traduzione diretta nel modello relazionale
4. ristrutturazione dello schema logico

Ristrutturazione dello schema ER

Motivazioni:

- **semplificare** la successiva fase di traduzione nel modello relazionale eliminando quei costrutti non direttamente traducibili
- tenere conto di aspetti relativi all'**efficienza**

Osservazione:

- uno schema ER **ristrutturato** è uno schema ER **degradato** dal punto di vista semantico per avvicinarsi al modello relazionale

Attività della ristrutturazione dello schema ER

1. analisi delle ridondanze
2. eliminazione degli attributi multivalore
3. eliminazione degli attributi composti
4. eliminazione delle ISA e delle generalizzazioni
5. scelta degli identificatori principali
6. specifica degli ulteriori vincoli esterni
7. riformulazione delle operazioni e delle specifiche sul carico applicativo in termini dello schema ristrutturato

Ristrutturazione – fase 1: analisi delle ridondanze

- Una **ridondanza** (estensionale) in uno schema ER è una informazione significativa ma derivabile da altre.
- Le ridondanze, se presenti, devono essere documentate (ovvero espresse attraverso **vincoli**).
- In questa fase si decide se eliminare le ridondanze eventualmente presenti o mantenerle, e se introdurne delle nuove.
- **Vantaggi** nel mantenere una ridondanza:
 - potenziale maggiore efficienza nella esecuzione delle interrogazioni
- **Svantaggi** nel mantenere una ridondanza:
 - gestione dei vincoli aggiuntivi
 - appesantimento degli aggiornamenti
 - maggiore occupazione di spazio

Analisi delle ridondanze

Abbiamo visto che le forme di ridondanza estensionale in uno schema ER sono date da:

- attributi derivabili:
 - da altri attributi della stessa entità (o relazione)
 - da attributi di altre entità (o relazioni)
- relazioni derivabili dalla composizione di altre relazioni in presenza di cicli

Per ciascuna ridondanza bisogna valutare, in funzione del carico applicativo previsto (aggiornamenti, interrogazioni, occupazione di spazio) se è opportuno mantenerla oppure eliminarla.

Se si sceglie di mantenere la ridondanza, questa deve essere **documentata** (attraverso opportuni vincoli).

Esempio di analisi delle ridondanze



Tabella dei volumi

| Concetto | Costrutto | Volume |
|-----------|-----------|-----------|
| Città | Entità | 200 |
| Persona | Entità | 1.000.000 |
| Residenza | Relazione | 1.000.000 |

Operazione 1: memorizza una nuova persona con la relativa città di residenza (500 volte al giorno)

Operazione 2: stampa tutti i dati di una città, incluso il numero di abitanti (2 volte al giorno)

Valutazione dei costi in assenza di ridondanza

Tabella degli accessi operazione 1

| Concetto | Costrutto | Accessi | Tipo |
|-----------|-----------|---------|------|
| Persona | Entità | 1 | S |
| Residenza | Relazione | 1 | S |

Tabella degli accessi operazione 2

| Concetto | Costrutto | Accessi | Tipo |
|-----------|-----------|---------|------|
| Città | Entità | 1 | L |
| Residenza | Relazione | 5000 | L |

Costi:

- operazione 1: 1000 accessi in scrittura al giorno
- operazione 2: 10000 accessi in lettura al giorno

Contiamo doppi gli accessi in scrittura: **totale di 12000 accessi al giorno**

Valutazione dei costi in presenza di ridondanza

Tabella degli accessi operazione 1

| Concetto | Costrutto | Accessi | Tipo |
|-----------|-----------|---------|------|
| Persona | Entità | 1 | S |
| Residenza | Relazione | 1 | S |
| Città | Entità | 1 | L |
| Città | Entità | 1 | S |

(scrittura dati su persona)
(associazione città di res.)
(lettura numero abitanti)
(scrittura nuovo num. abit.)

Tabella degli accessi operazione 2

| Concetto | Costrutto | Accessi | Tipo |
|----------|-----------|---------|------|
| Città | Entità | 1 | L |

Costi:

- operazione 1: 1500 accessi in scrittura e 500 in lettura al giorno
- operazione 2: trascurabile

Contiamo doppi gli accessi in scrittura: **totale di 3500 accessi al giorno**

Conclusione: in questo esempio, manteniamo la ridondanza

Ristrutturazione – fase 2: eliminazione di attributi multivalore

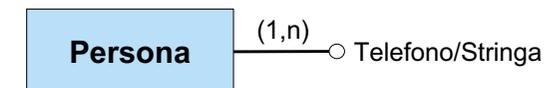
- Un **attributo multivalore** (ovvero un attributo con cardinalità massima maggiore di 1) non può essere tradotto direttamente nel modello relazionale senza introdurre delle ridondanze nelle relazioni ottenute.
- Dobbiamo quindi eliminare tutti gli attributi multivalore.
- L'eliminazione di un **attributo multivalore di un'entità** si effettua trasformando l'attributo in una relazione binaria, ed introducendo un'opportuna entità per il dominio (cfr. parte 4, esercizio 19a).
- L'eliminazione di un **attributo multivalore di una relazione** richiede la preventiva trasformazione della relazione in un'entità (cfr. parte 4, esercizio 19c e 19d).

Nota: se trasformiamo una relazione R in un'entità, e R stava in ISA o in una gerarchia con altre relazioni, allora anche queste devono essere trasformate in entità (cfr. parte 4 esercizio 18c con ISA tra relazioni).

Eliminazione di attributi multivalore di entità

Si trasforma l'attributo multivalore dell'entità in una relazione e il corrispondente dominio in entità.

Esempio:

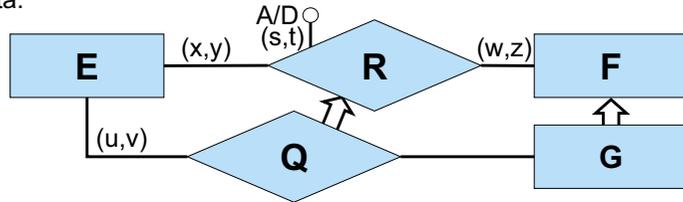


Si trasforma in:

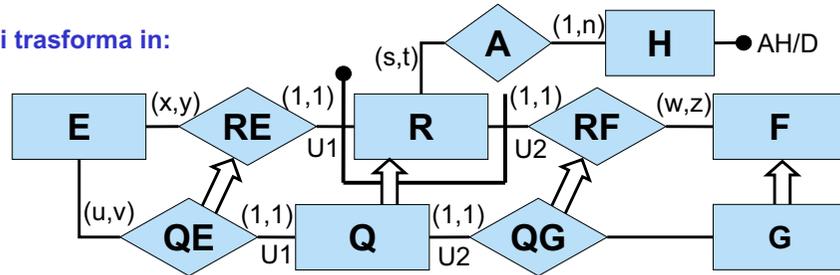


Eliminazione di attributi multivalore di relazioni

Si trasforma la relazione R in entità e l'attributo multivalore di R in una relazione. Anche eventuali relazioni in ISA con R devono essere trasformate in entità.



Si trasforma in:



Ristrutturazione – fase 3: eliminazione di attributi composti

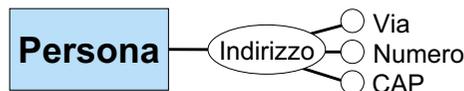
Un **attributo composto** di un'entità (o di una relazione) a questo punto ha cardinalità (1,1) oppure (0,1).

- Se l'attributo ha cardinalità (1,1), si associano direttamente gli attributi componenti all'entità (o alla relazione).
 - Se l'attributo ha cardinalità (0,1) si può
 - procedere come per gli attributi (1,1), ma con l'avvertenza che l'opzionalità diventa un vincolo esterno
 - oppure trasformare l'attributo composto in una relazione binaria introducendo una nuova entità, come fatto per gli attributi multivalore (cfr. parte 4 esercizio 19b), mantenendo la cardinalità (0,1) sulla relazione.
- Ovviamente, se l'attributo composto è di una relazione, è necessario trasformare tale relazione in un'entità.

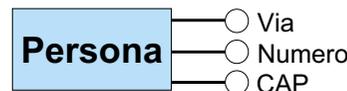
Eliminazione di attributi composti: alternativa 1

Si associano direttamente gli attributi componenti all'entità (o alla relazione) a cui è associato l'attributo:

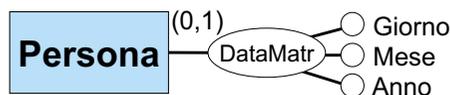
Esempio:



viene trasformato in:



Esempio:



viene trasformato in:

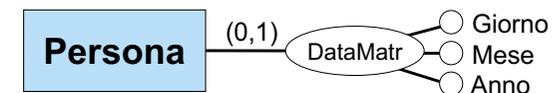


... con il vincolo esterno: *per ogni istanza di Persona, ciascun attributo tra GMatr, MMatr e AMatr è definito se e solo se lo sono anche gli altri due.*

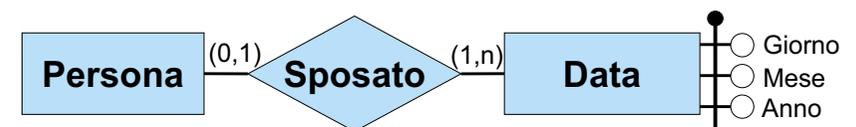
Eliminazione di attributi composti: alternativa 2

Si trasforma l'attributo composto in una relazione binaria e si introduce una nuova entità.

Esempio:



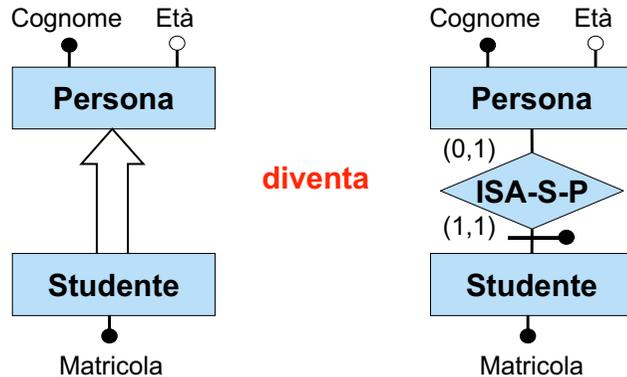
Viene trasformato in:



Ristrutturazione – fase 4: eliminazione di ISA tra entità

Una relazione **E ISA F** tra due entità **E** ed **F** viene sostituita da una nuova relazione binaria **ISA-E-F** tra **E** ed **F** a cui **E** partecipa con cardinalità (1,1) e **F** con cardinalità (0,1). Agli eventuali identificatori di **E** viene aggiunto un identificatore esterno dato dalla partecipazione ad **ISA-E-F**.

Esempio:

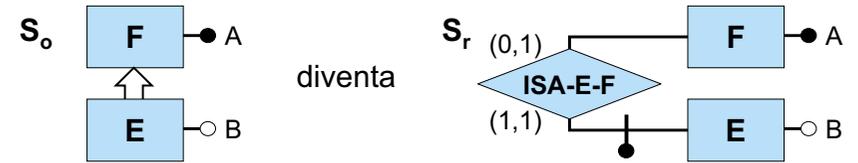


Giuseppe De Giacomo

Basi di Dati

Progettazione logica - 29

Eliminazione di ISA tra entità: livello estensionale



Istanza di S₀:

istanze(F) = { f₁, f₂, f₃ }
 istanze(E) = { e₁, e₂ }
 istanze(A) = { (f₁, a₁), (f₂, a₂), (f₃, a₃) }
 istanze(B) = { (f₁, b₁), (f₂, b₂) }

Istanza di S_r:

istanze(F) = { f₁, f₂, f₃ }
 istanze(E) = { e₁, e₂ }
 istanze(A) = { (f₁, a₁), (f₂, a₂), (f₃, a₃) }
 istanze(B) = { (e₁, b₁), (e₂, b₂) }
 istanze(ISA-E-F) = { (E:e₁, F:f₁), (E:e₂, F:f₂) }

Si noti che nello schema S_r risultante dall'eliminazione delle ISA da S₀, **tutte le entità sono disgiunte a coppie**, e quindi non hanno più istanze in comune (su questo torneremo fra breve).

Giuseppe De Giacomo

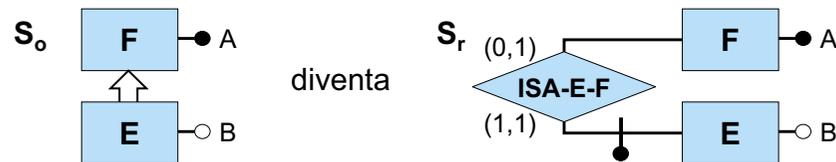
Basi di Dati

Progettazione logica - 30

Eliminazione di ISA tra entità: corrispondenza tra istanze

Esiste una stretta corrispondenza tra le istanze di uno schema S₀ e le istanze dello schema S_r ottenuto da S₀ eliminando le ISA tra entità.

Mostriamo questa proprietà tramite un esempio:



In particolare, mostriamo che esistono due funzioni **g** ed **h** tali che:

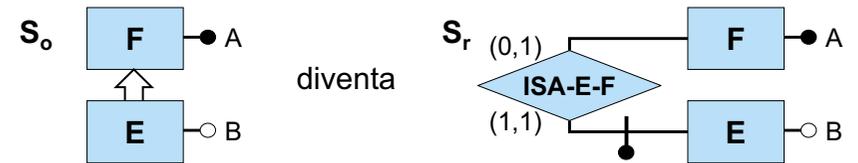
- **g** è una funzione totale da *istanze*(S₀) a *istanze*(S_r)
- **h** è una funzione totale da *istanze*(S_r) a *istanze*(S₀)
- per ogni istanza *l₀* di S₀, si ha che **h(g(l₀)) = l₀**

Giuseppe De Giacomo

Basi di Dati

Progettazione logica - 31

Corrispondenza tra istanze – funzione g



Definiamo la funzione **g**: *istanze*(S₀) → *istanze*(S_r) in modo che, ad un'istanza *l₀* di S₀, **g** assegni un'istanza *l_r* di S_r definita al seguente modo:

- *istanze*(l_r, F) = *istanze*(l₀, F)
- *istanze*(l_r, A) = *istanze*(l₀, A)
- per definire *istanze*(l_r, E), introduciamo in l_r, per ogni *x* ∈ *istanze*(l₀, E), un nuovo oggetto **g_E(x)**, e definiamo
istanze(l_r, E) = { g_E(x) | x ∈ *istanze*(l₀, E) }
- *istanze*(l_r, ISA-E-F) = { (g_E(x), x) | x ∈ *istanze*(l₀, E) }
- *istanze*(l_r, B) = { (g_E(x), b) | (x, b) ∈ *istanze*(l₀, B) }

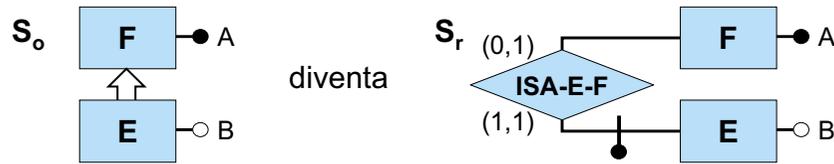
È facile verificare che l_r così definita è effettivamente un'istanza di S_r.

Giuseppe De Giacomo

Basi di Dati

Progettazione logica - 32

Corrispondenza tra istanze – funzione h

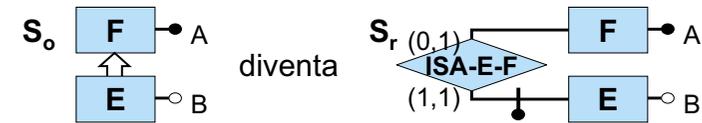


Definiamo la funzione h : $istanze(S_r) \rightarrow istanze(S_o)$ in modo che, ad un'istanza l_r di S_r , h assegni l'istanza l_o di S_o definita al seguente modo:

- $istanze(l_o, F) = istanze(l_r, F)$
- $istanze(l_o, A) = istanze(l_r, A)$
- $istanze(l_o, E) = \{ x \in istanze(l_r, E) \mid \text{esiste un } y \in istanze(l_r, F) \text{ con } (y, x) \in istanze(l_r, ISA-E-F) \}$
- $istanze(l_o, B) = \{ (x, b) \mid x \in istanze(l_o, E), (y, x) \in istanze(l_r, ISA-E-F) \text{ e } (y, b) \in istanze(l_r, E) \}$

È facile verificare che l_o così definita è effettivamente un'istanza di S_o , e che inoltre $h(g(l_o)) = l_o$.

Osservazione sullo schema risultante



Istanza di S_o :

$istanze(F) = \{ f_1, f_2, f_3 \}$

$istanze(E) = \{ e_1, e_2 \}$

$istanze(A) = \{ (f_1, a_1), (f_2, a_2), (f_3, a_3) \}$

$istanze(B) = \{ (f_1, b_1), (f_2, b_2) \}$

Istanza di S_r :

$istanze(F) = \{ f_1, f_2, f_3 \}$

$istanze(E) = \{ e_1, e_2 \}$

$istanze(A) = \{ (f_1, a_1), (f_2, a_2), (f_3, a_3) \}$

$istanze(B) = \{ (e_1, b_1), (e_2, b_2) \}$

$istanze(ISA-E-F) = \{ (E:e_1, F:f_1), (E:e_2, F:f_2) \}$

Come osservato prima, nello schema S_r risultante dall'eliminazione delle ISA da S_o , **tutte le entità sono disgiunte a coppie**.

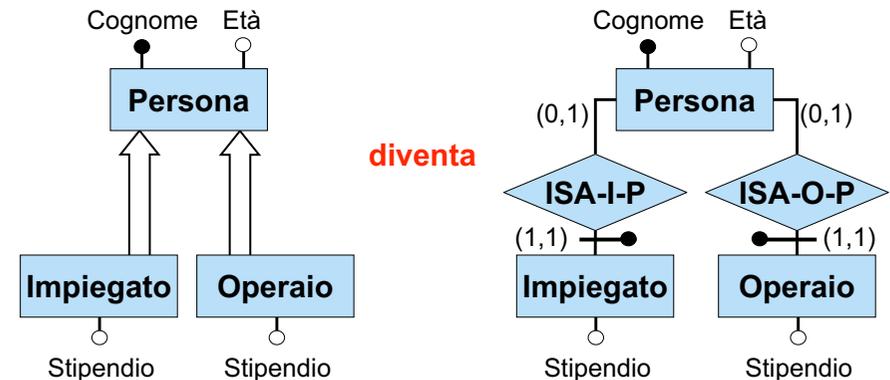
Come si concilia questa osservazione con il fatto che nello schema originario ciò non era vero? La risposta sta nelle funzioni g ed h illustrate precedentemente. Tramite queste funzioni è possibile infatti stabilire se due qualunque istanze di entità nello schema risultante corrispondono ad un'unica istanza di entità nello schema originario (ad es., e_1 ed f_1 nella istanza di S_r corrispondono entrambi ad f_1 nella istanza di S_o).

Attributi in comune nella eliminazione di ISA tra entità

Se due entità non disgiunte nello schema originario hanno un attributo A in comune, applicando la trasformazione per eliminare la relazione ISA, nelle due corrispondenti entità disgiunte dello schema risultante troviamo due attributi di nome A , che di fatto rappresentano due funzioni (o relazioni, se A è multivalore) diverse.

Dobbiamo quindi imporre che le due funzioni, quando applicate a due oggetti e , f dello schema risultante che corrispondono allo stesso oggetto dello schema originario, assegnino ad e ed f lo stesso valore. Ciò viene fatto con un **vincolo esterno sullo schema ristrutturato**.

Attributi in comune nella eliminazione di ISA tra entità: esempio



con il **vincolo esterno** nello schema risultante:

per ogni $p \in istanze(Persona)$, se esistono $i \in istanze(Impiegato)$, e $o \in istanze(Operaio)$ tali che $(i, p) \in istanze(ISA-I-P)$ e $(o, p) \in istanze(ISA-O-P)$, allora $Stipendio(i) = Stipendio(o)$

Ristrutturazione – fase 4: eliminazione di generalizzazioni tra entità

- Una generalizzazione tra una entità padre **F** e le sottoentità **E₁, E₂, ..., E_n**, viene trattata come n relazioni **E₁ ISA F, ..., E_n ISA F**, introducendo n relazioni binarie **ISA-E₁-F, ..., ISA-E_n-F**.
- Per tenere conto delle proprietà delle generalizzazioni si aggiungono opportuni vincoli esterni, detti **vincoli di generalizzazione**:

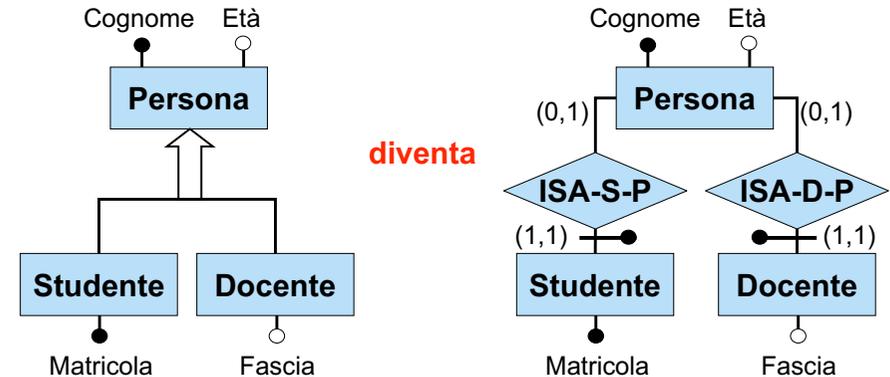
- la proprietà $istanze(E_1) \cap \dots \cap istanze(E_n) = \emptyset$ dello schema di partenza corrisponde nello schema ristrutturato al vincolo:

*ogni istanza di F partecipa
al più ad una delle relazioni ISA-E₁-F, ..., ISA-E_n-F*

- se la **generalizzazione è completa**, l'ulteriore proprietà $istanze(E_1) \cup \dots \cup istanze(E_n) = istanze(F)$ dello schema di partenza corrisponde nello schema ristrutturato al vincolo:

*ogni istanza di F partecipa
esattamente ad una delle relazioni ISA-E₁-F, ..., ISA-E_n-F*

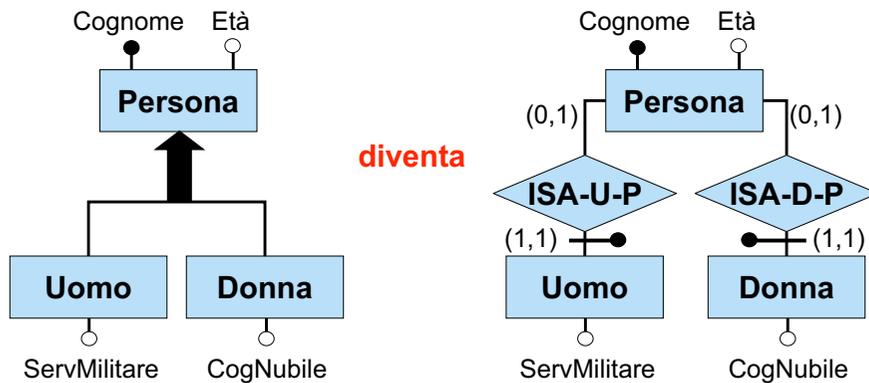
Eliminazione di generalizzazioni tra entità



Vincoli di generalizzazione:

*nessuna istanza di Persona partecipa sia a ISA-S-P
sia a ISA-D-P*

Eliminazione di generalizzazioni complete tra entità



Vincoli di generalizzazione:

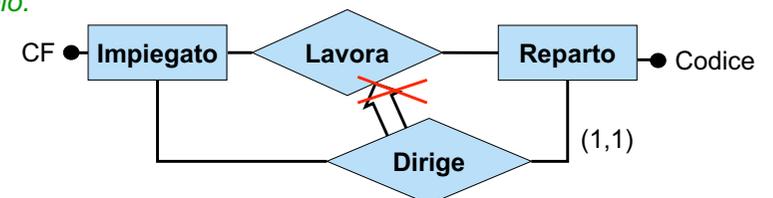
*ogni istanza di Persona partecipa ad ISA-U-P oppure
ad ISA-D-P, ma non ad entrambi*

Ristrutturazione – fase 4: eliminazione di ISA e generalizzazioni tra relazioni

Le **relazioni ISA e le generalizzazioni tra relazioni** vengono eliminate dallo schema e vengono espresse tramite opportuni vincoli esterni.

Nel caso in cui le relazioni in ISA (o nella generalizzazione) insistano su esattamente le **stesse entità per tutti i ruoli**, è immediato esprimere il vincolo esterno.

Esempio:

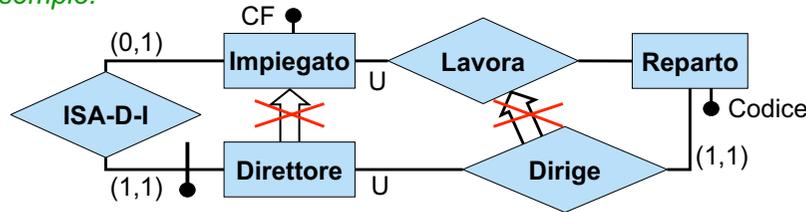


Vincolo esterno: ogni istanza di Dirige è anche un'istanza di Lavora.

Eliminazione di ISA e generalizzazioni tra relazioni

Nel caso in cui le relazioni in ISA (o nella generalizzazione) insistono su **entità diverse in qualche ruolo**, nell'esprimere il vincolo esterno bisogna tenere conto che nello schema ristrutturato entità diverse sono tra loro disgiunte.

Esempio:



Vincolo esterno: per ogni istanza $(U:d, Reparto:r)$ di Dirige, sia i l'istanza di Impiegato tale che $(Direttore:d, Impiegato:i)$ è un'istanza di ISA-D-I (si noti che i esiste sempre ed è unica); allora $(U:i, Reparto:r)$ deve essere un'istanza di Lavora.

Ristrutturazione – fase 5: scelta degli identificatori principali

Per **ogni entità** è necessario:

- individuare almeno un identificatore
- scegliere tra gli identificatori dell'entità un **identificatore principale**.

Criteri per la scelta dell'identificatore principale:

- semplicità (cioè con pochi campi)
- preferenza per gli identificatori interni
- utilizzo nelle operazioni più frequenti o importanti
- se per un'entità nessuno degli identificatori soddisfa tali requisiti, è possibile introdurre un ulteriore attributo dell'entità (un **codice**, i cui valori sono speciali ed hanno l'unico scopo di identificare le istanze dell'entità).

In una entità con più identificatori, quello principale viene indicato nella documentazione associata allo schema ristrutturato. Sulle slide, in presenza di più identificatori per un'entità, denoteremo quello principale con un cerchio aggiuntivo.

Cicli di identificazione esterna

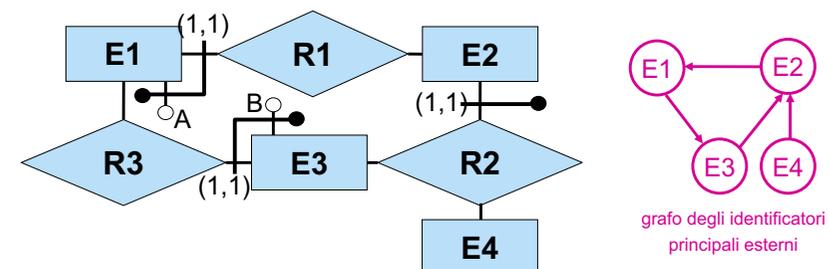
Nella scelta degli identificatori principali è necessario fare attenzione a non introdurre **cicli di identificazione esterna**.

Definiamo il **grafo degli identificatori (principali) esterni** nel seguente modo:

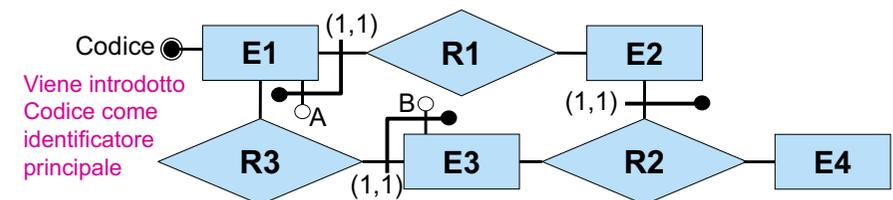
- ad ogni entità del diagramma corrisponde un nodo
- c'è un arco dall'entità E all'entità F se E partecipa ad una relazione che è parte dell'identificatore (principale) esterno di F.

Si ha un ciclo di identificazione esterna quando il grafo degli identificatori principali esterni contiene un ciclo.

Cicli di identificazione esterna



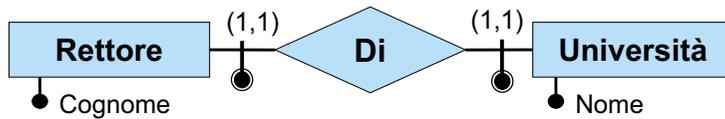
È necessario **spezzare i cicli di identificazione esterna** scegliendo per almeno una entità nel ciclo un identificatore principale diverso. Se non ci sono alternative, è necessario introdurre un opportuno codice.



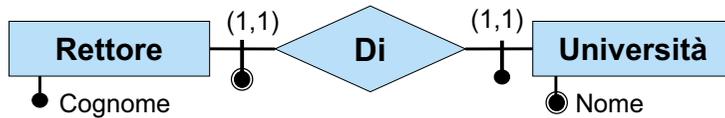
Cicli di identificazione esterna: esempio

Un caso significativo di ciclo di identificazione esterna è dato da due entità che si identificano a vicenda.

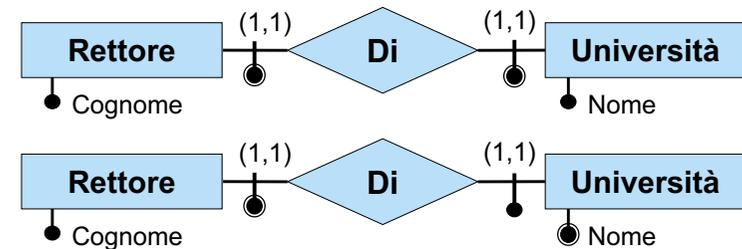
Esempio:



Abbiamo un ciclo di identificazione esterna, che deve essere spezzato. Una possibilità è la seguente:

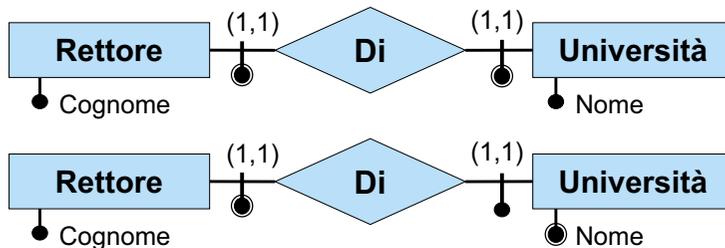


Esercizio 2: cicli di identificazione esterna



- Che differenza c'è tra le istanze dei due schemi?
- Perché è necessario spezzare i cicli di identificazione esterna?

Esercizio 2: soluzione



I due schemi hanno le stesse istanze, in quanto la scelta degli identificatori principali non ha portato all'introduzione di nuovi attributi.

I cicli di identificazione esterna non rappresentano alcun problema per quanto riguarda lo schema concettuale.

È però necessario spezzarli perché renderebbero impossibile la traduzione nel modello relazionale. Vedremo infatti che, se un'entità E ha un identificatore principale esterno su una relazione R, nello schema relazionale prodotto, ad E corrisponderà una relazione la cui chiave è data dagli identificatori di tutte le entità che partecipano ad R. In presenza di cicli di identificazione esterna questo accorpamento non è fattibile.

Ristrutturazione – fase 6: specifica degli ulteriori vincoli esterni

- È necessario riformulare tutti i vincoli esterni dello schema originario in termini dello schema ristrutturato.
 - mettere in evidenza anche i vincoli impliciti (ovvero che sono conseguenza di altri vincoli)
Es. vincoli di identificazione esterna per relazioni (1,1) – (1,1) oppure (1,1) – (0,1)
- Si devono aggiungere i vincoli derivanti dalla ristrutturazione:
 - vincoli derivanti da attributi composti opzionali
 - vincoli per due entità che erano in ISA con una stessa entità padre e che hanno attributi in comune
 - vincoli di generalizzazione (disgiuntezza e completezza)
 - vincoli dovuti agli identificatori non principali (se non sono più rappresentati nello schema)

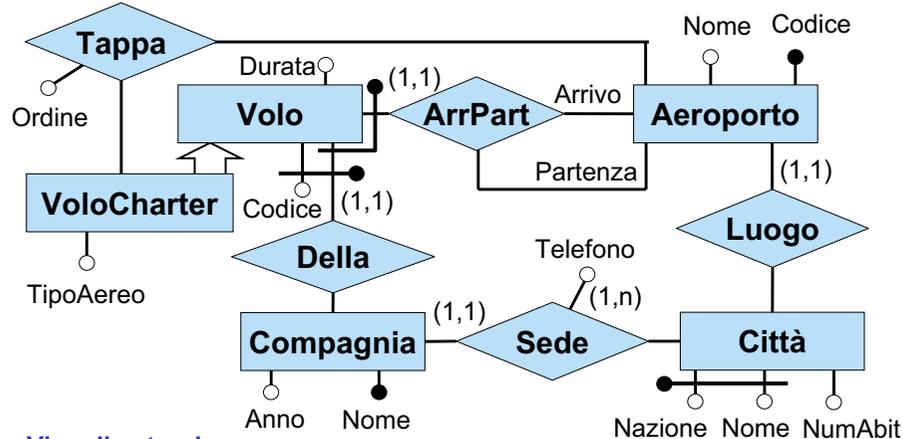
Ristrutturazione – fase 7: riformulazione di operazioni e carico applicativo

- È necessario riformulare le operazioni e i relativi schemi di navigazione in termini dello schema ristrutturato.
- È necessario riformulare le specifiche sul carico applicativo in termini dello schema ristrutturato.

Riassunto sulla ristrutturazione

1. Analisi delle ridondanze (si tiene conto dell'efficienza)
2. Eliminazione degli attributi multivalore
3. Eliminazione degli attributi composti (eventuale vincolo (0,1) diventa vincolo esterno)
4. Eliminazione delle ISA e delle generalizzazioni
 - vincoli per entità figlie della stessa entità padre con uno stesso attributo
 - vincoli di generalizzazione (disgiuntezza e completezza)
 - si noti che tutte le entità diventano disgiunte
5. Scelta degli identificatori principali
 - tutte le entità devono avere un identificatore (eventualmente, introdurre codice)
 - eliminazione di cicli di identificatori principali esterni
6. Specifica degli ulteriori vincoli esterni
 - vincoli derivanti dalla ristrutturazione
 - riformulazione dei vincoli esterni dello schema originario
7. Riformulazione delle operazioni e delle specifiche sul carico applicativo in termini dello schema ristrutturato

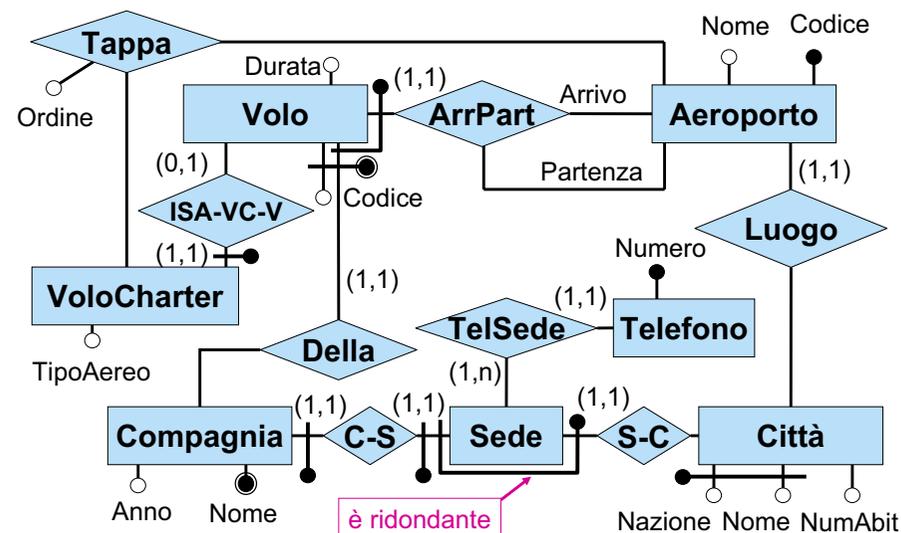
Esercizio 3: ristrutturare il seguente schema



Vincoli esterni:

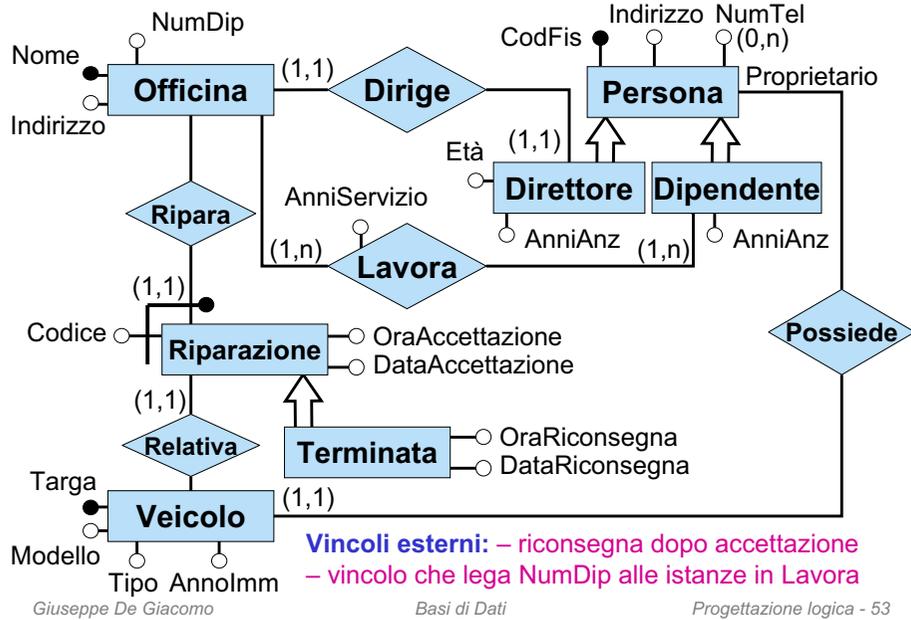
- 1) per ogni v in VoloCharter, se $(v, a_1), \dots, (v, a_n)$ sono tutte le coppie in Tappa alle quali partecipa v , e se o_1, \dots, o_n sono i valori assegnati a tali coppie dall'attributo Ordine, allora per $i=1, \dots, n$ esiste un o_i tale che $o_i = i$.
- 2) Un telefono è di una sola sede.

Esercizio 3: soluzione



Vincolo esterno: 1) vincolo su Ordine in Tappa (2 è diventato interno allo schema)

Esercizio 4: ristrutturare il seguente schema



Esercizio 4: soluzione

