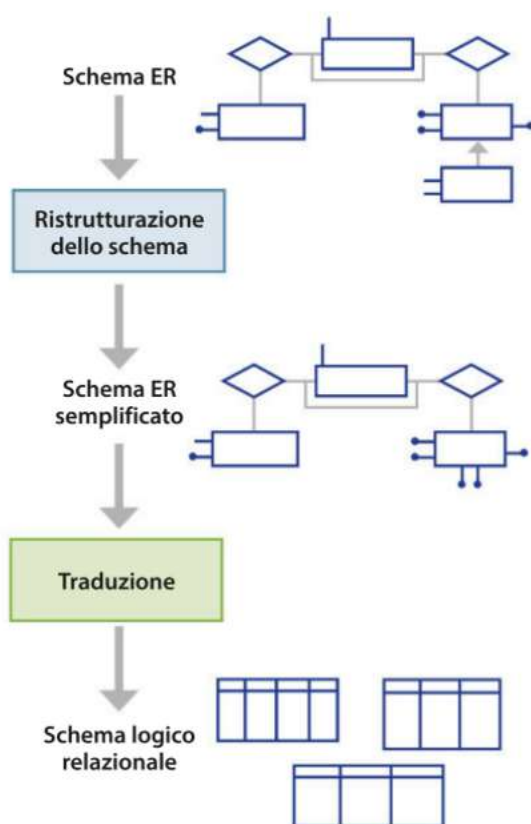


# La progettazione logica: il modello relazionale

## In questa unità imparerai...

- A che cosa serve la progettazione logica
- Come tradurre uno schema concettuale in uno schema logico
- Quali sono e come si usano gli operatori dell'algebra relazionale
- Come normalizzare le relazioni

## 1 La progettazione logica



L'obiettivo di questa fase è produrre uno schema logico in grado di descrivere, in modo corretto ed efficiente, tutte le informazioni rappresentate dallo schema ER prodotto nella fase di progettazione concettuale. Non si tratta però di una semplice trasformazione da uno schema a un altro, in quanto, prima di passare allo schema logico, il diagramma ER deve essere ristrutturato in modo da **semplificare** la traduzione e **ottimizzare** il progetto finale.

La semplificazione si rende necessaria in quanto non tutti i costrutti dello schema ER hanno la rispettiva traduzione nel modello logico. È utile articolare la progettazione logica in due fasi:

**ristrutturazione dello schema Entità-Associazione:** è una fase indipendente dal modello logico scelto e si concentra sui criteri di ottimizzazione dello schema e di semplificazione per la fase successiva;

**traduzione nel modello logico:** si tiene conto di uno specifico modello logico (nel nostro caso quello relazionale) all'interno del quale viene tradotta la realtà rappresentata dallo schema ER.

Il risultato ottenuto dopo la fase di ristrutturazione non è più uno schema concettuale nel senso stretto del termine, in quanto tiene conto anche degli aspetti realizzativi.

## 2 Ristrutturazione dello schema ER

Il seguente elenco rappresenta l'insieme dei passi da svolgere, in sequenza, nella fase di ristrutturazione dello schema Entità-Associazioni.

- **Analisi delle ridondanze:** vengono individuate eventuali ridondanze per stabilire se è necessario eliminarle o mantenerle.
- **Eliminazione delle generalizzazioni:** tutte le generalizzazioni vengono sostituite con altri costrutti del modello ER.
- **Partizionamento/accorpamento di entità e associazioni:** si valuta se è opportuno separare (partizionare) o aggregare (accorpare) concetti presenti nello schema.
- **Scelta delle chiavi primarie:** la scelta delle chiavi primarie è essenziale nella traduzione verso lo schema relazionale, perché le chiavi che identificano le occorrenze consentono di stabilire i legami tra i dati.

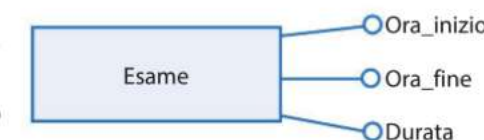
### Analisi delle ridondanze

Una **ridondanza** in uno schema ER è una informazione significativa ma derivabile da altre. Di norma le ridondanze andrebbero eliminate, ma occorre tenere conto di vari fattori da analizzare.

Riassumiamo di seguito i differenti casi di ridondanza che si possono presentare.

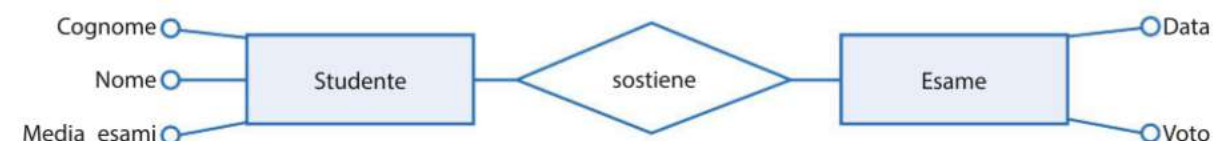
#### Attributi derivabili all'interno della stessa entità

Questo tipo di ridondanza deve essere eliminata rimuovendo gli attributi che possono essere dedotti da altri attraverso dei calcoli. La durata dell'esame verrà ricavata dalla differenza tra l'ora di inizio e di fine.



#### Attributi derivabili da altre entità

L'associazione fra due entità può portare a rendere ridondante qualche attributo; anche in questo caso, la stessa informazione può essere dedotta a partire da altre.



La media degli esami, per ogni studente, sarà calcolata in base a tutti i voti degli esami sostenuti dal singolo studente.

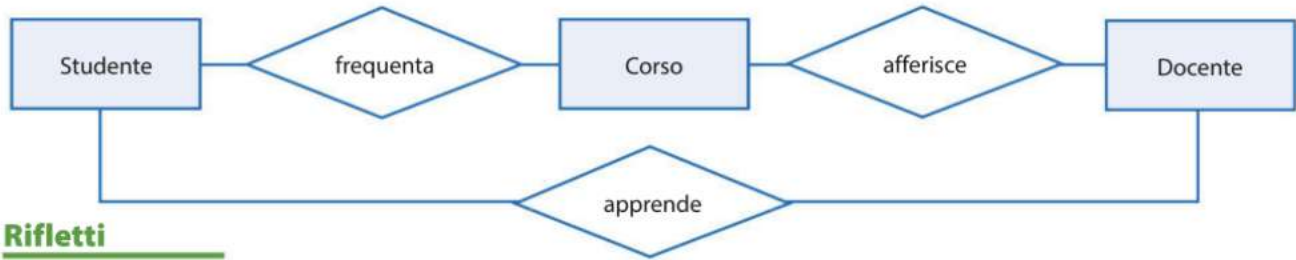
#### Attributi derivabili da operazioni di conteggio di occorrenze

Ad esempio, un attributo *Numero Abitanti* per un'entità *Città* potrebbe essere desunto dal conteggio degli abitanti residenti; si tratta in effetti di una variante del caso precedente, che viene però discussa separatamente in quanto si verifica molto di frequente.

#### Ridondanza di associazioni in presenza di cicli

Un uso eccessivo di associazioni può portare ad avere dei *cicli*, cioè dei percorsi alternativi per raggiungere la stessa informazione.





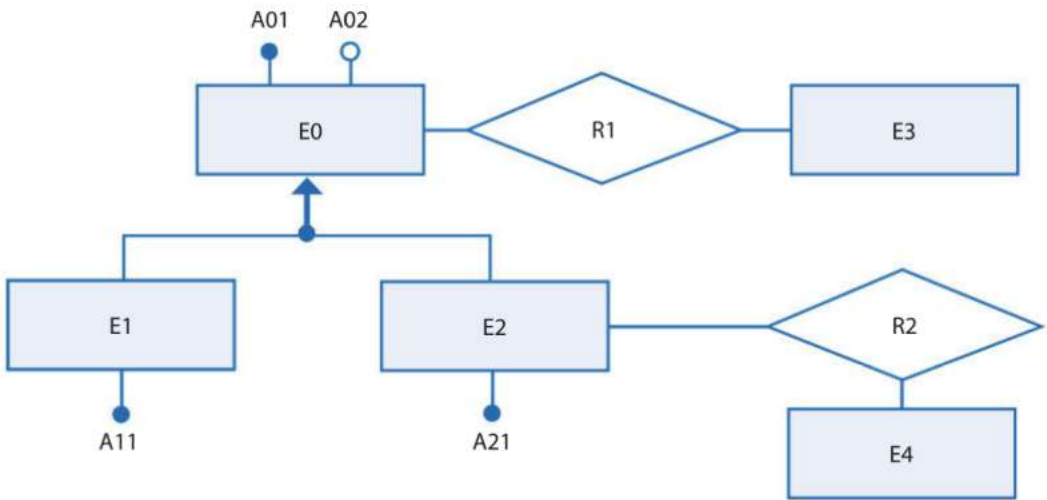
Rifletti

La presenza di ridondanze non è sempre uno svantaggio, anche se implica una occupazione di memoria aggiuntiva per rappresentare informazioni già presenti. La decisione di mantenere una ridondanza va presa confrontando il costo di esecuzione delle operazioni per l'informazione ridondante e la relativa occupazione di memoria; il vantaggio consiste in una riduzione degli accessi rispetto a quelli necessari per calcolare il dato derivato. Nel caso qui trattato, supponiamo che il sistema debba estrarre molte volte al giorno un elenco aggiornato degli studenti ripartiti per docente di ogni singolo corso; senza la ridondanza, il passaggio per le altre due associazioni sarebbe necessario, e se la mole dei dati presenti è rilevante, l'operazione richiesta deve scomodare tutti i dati relativi ai docenti, ai corsi e agli studenti. Pertanto, nel caso in cui l'operazione in questione dovesse essere eseguita molte volte al giorno su un quantitativo di dati notevole, sarebbe auspicabile mantenere l'associazione ridondante.

In questo caso l'associazione *apprende* può considerarsi superflua in quanto le associazioni *afferisce* e *frequenta* realizzano già il collegamento fra *Studente* e *Docente* (il docente insegna in un corso frequentato da studenti, di contro anche lo studente frequenta un corso tenuto da uno specifico docente), pertanto l'informazione diretta relativa all'associazione studente-docente può essere dedotta indirettamente dalle altre associazioni presenti.

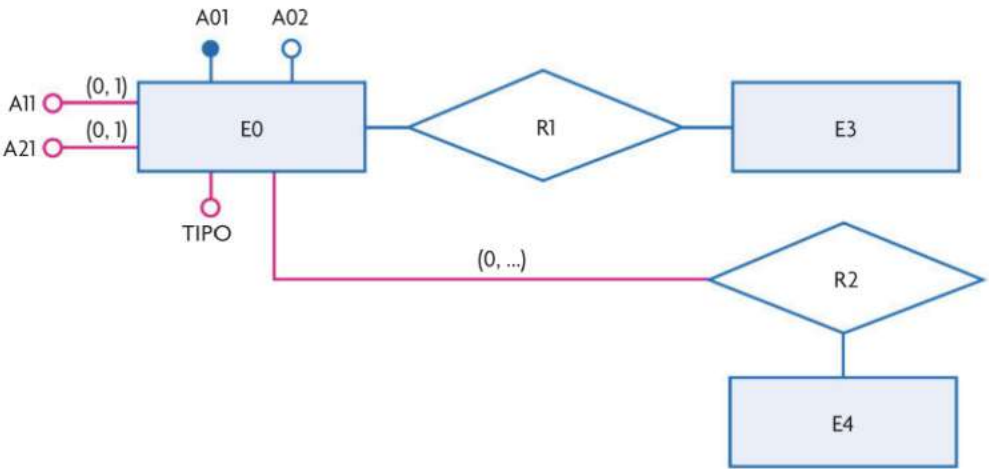
Eliminazione delle generalizzazioni

I modelli logici tradizionali, tra cui il modello relazionale, non consentono di rappresentare direttamente una generalizzazione, pertanto è necessario trasformare questo costruito in altri costrutti del modello ER che si basano esclusivamente su entità e associazioni. Per fare ciò abbiamo a disposizione tre alternative possibili. Facendo riferimento al seguente schema ER osserviamo come sia possibile risolvere la generalizzazione attraverso le differenti strategie di ristrutturazione.



1. Accorpamento delle figlie della generalizzazione nel padre.

Le entità E1 e E2 vengono eliminate e i loro attributi, che divengono opzionali, vengono aggiunti a quelli dell'entità padre E0; a tale entità va inoltre aggiunto un attributo che serve a distinguere il tipo, cioè se l'occorrenza di E0 apparteneva a E1 o E2. Ad esempio, una generalizzazione tra l'entità *Persona* e le entità *Uomo* e *Donna* viene ristrutturata aggiungendo all'entità *Persona* l'attributo *Sesso*. Tale soluzione conviene quando gli accessi all'entità padre e alle entità figlie sono contestuali.

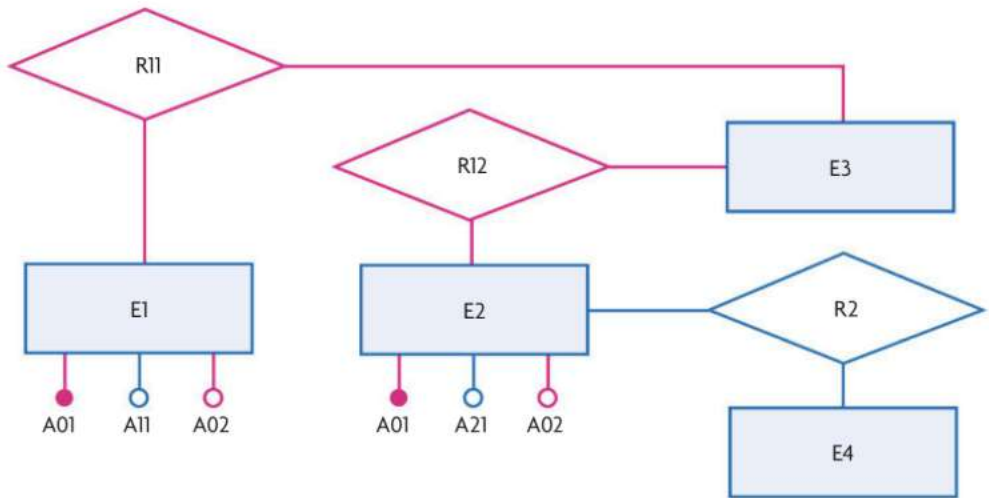


Rifletti

Nel caso in cui la generalizzazione sia non totale, l'attributo TIPO avrà il valore null e, anche se si ha uno spreco di memoria per la presenza di tali valori null, questa soluzione assicura un minor numero di accessi.

2. Accorpamento del padre della generalizzazione nelle figlie.

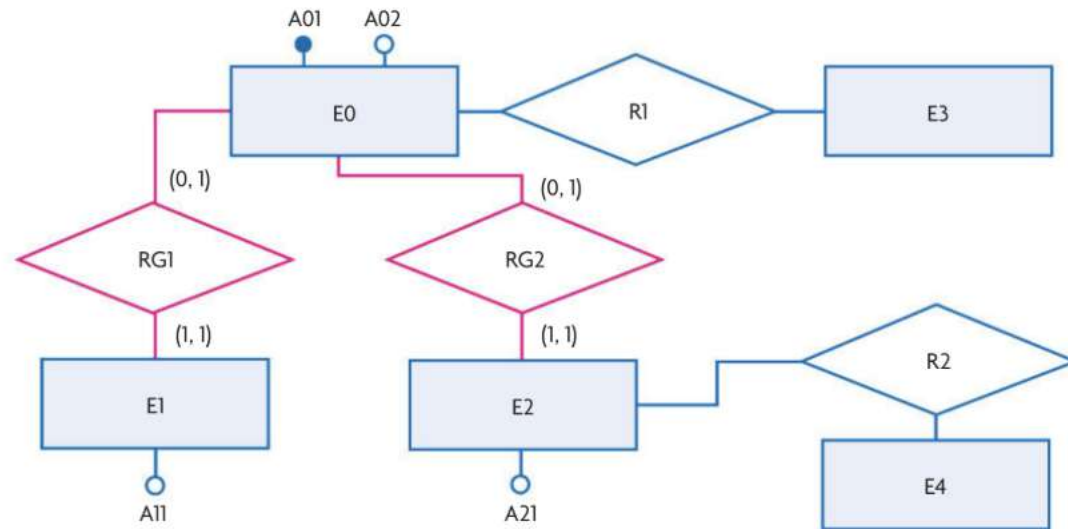
L'entità padre E0 viene eliminata e i suoi attributi trasferiti a entrambe le figlie; l'associazione R1 viene scissa in due associazioni distinte R11 e R12. Questo tipo di ristrutturazione può essere impiegato quando la generalizzazione è totale, altrimenti le occorrenze di E0 che non sono occorrenze né di E1 né di E2 non sarebbero rappresentate. È conveniente quando ci sono operazioni che si riferiscono solo a occorrenze di E1 oppure di E2, facendo una netta distinzione fra le entità. La scelta garantisce un risparmio di memoria rispetto alla soluzione n. 1 per la mancanza dei valori nulli, e una riduzione degli accessi rispetto alla soluzione successiva (n. 3), in quanto non occorre visitare E0 per raggiungere gli attributi delle figlie.



3. Sostituzione della generalizzazione con associazioni.

La generalizzazione si trasforma in due associazioni uno a uno che legano l'entità padre con le figlie, non sono previsti trasferimenti di attributi con le entità E1 e E2 e sono identificate esternamente dall'entità E0. Tale soluzione conviene quando la generalizzazione non è totale e quando gli accessi alle entità figlie sono separati dagli accessi al padre. In questo caso vi è un risparmio di memoria rispetto alla scelta n.1 per l'assenza dei valori nulli, ma un maggior numero di accessi per garantire la consistenza dei dati.





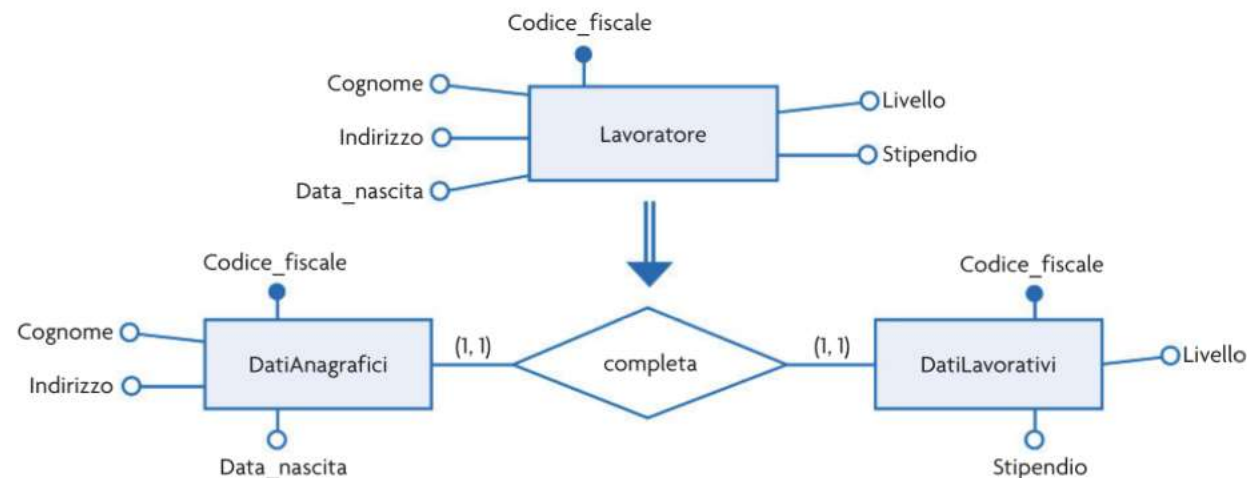
La ristrutturazione delle generalizzazioni, tra le varie alternative proposte, può essere svolta in modo analogo a quanto previsto per gli attributi derivati, considerando vantaggi e svantaggi di ognuna delle scelte possibili relativamente all'occupazione di memoria e al costo delle operazioni coinvolte. Dalle scelte esaminate sembrerebbe che la n. 3 sia quella più onerosa in termini di accessi e quindi non dovrebbe essere presa in considerazione, di fatto però questa soluzione ha il vantaggio di produrre entità con pochi attributi e quindi strutture logiche di piccole dimensioni, il cui accesso garantirebbe un numero elevato di dati prelevati.

### Partizionamento/accorpamento di entità e associazioni

Le entità e le relazioni di uno schema ER possono essere partizionate o accorpate con lo scopo di ottenere una maggiore efficienza delle operazioni, ossia di ridurre il numero di accessi mediante separazione di attributi dello stesso concetto o raggruppando attributi di concetti differenti.

#### Partizionamento di entità

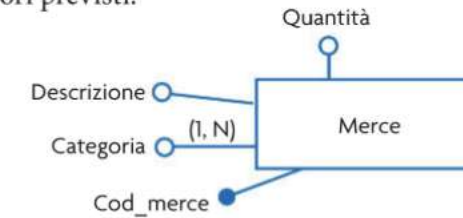
Un esempio di partizionamento di entità è mostrato nella figura seguente; l'entità *Lavoratore* viene sostituita da due entità collegate da un'associazione uno a uno che descrivono, rispettivamente, i dati anagrafici dei lavoratori e i dati relativi alla loro retribuzione.



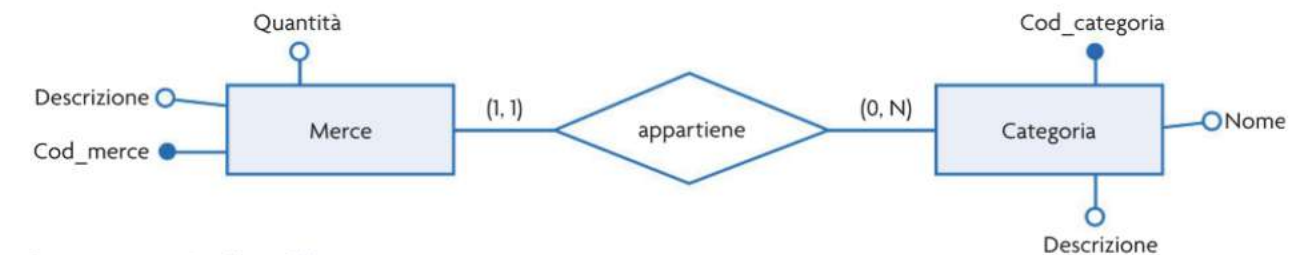
Un partizionamento di questo tipo è un esempio di **decomposizione verticale** di un'entità, poiché si suddivide il concetto operando sui suoi attributi. Tale ristrutturazione ha senso se le operazioni che coinvolgono di frequente le **entità originarie** richiedono, per il lavoratore, solo informazioni di natura anagrafica o solo dati relativi alla sua retribuzione. Una **decomposizione orizzontale**, invece, viene realizzata sulle occorrenze dell'entità, ad esempio il lavoratore potrebbe essere un impiegato o un operaio; così facendo le entità ottenute avranno gli stessi attributi dell'entità di partenza. Di fatto, una decomposizione orizzontale corrisponde a una generalizzazione e presenta lo svantaggio di duplicare tutte le associazioni a cui l'entità originaria è legata.

#### Eliminazione di attributi multivalore

Come abbiamo già visto nell'unità didattica precedente, in questo esempio notiamo la presenza di un attributo multivalore (*Categoria*), cioè di un attributo che può contenere uno dei particolari valori previsti.



La ristrutturazione si rende necessaria perché il modello relazionale non permette di rappresentare questo tipo di attributi. La ristrutturazione è molto semplice e consiste nel realizzare un'associazione 1 a N tra l'entità originaria e una nuova entità che prende il nome dell'attributo multivalore; eventuali nuovi attributi possono essere inseriti nella nuova entità generata per rendere il tutto più chiaro.



#### Accorpamento di entità

L'accorpamento è l'operazione inversa del partizionamento e si effettua in genere su associazioni di tipo uno a uno. Questo tipo di ristrutturazione può essere suggerito dal fatto che le operazioni più frequenti su una determinata entità richiedono sempre i dati relativi all'entità potenzialmente da accorpere.

