

23-11-2022

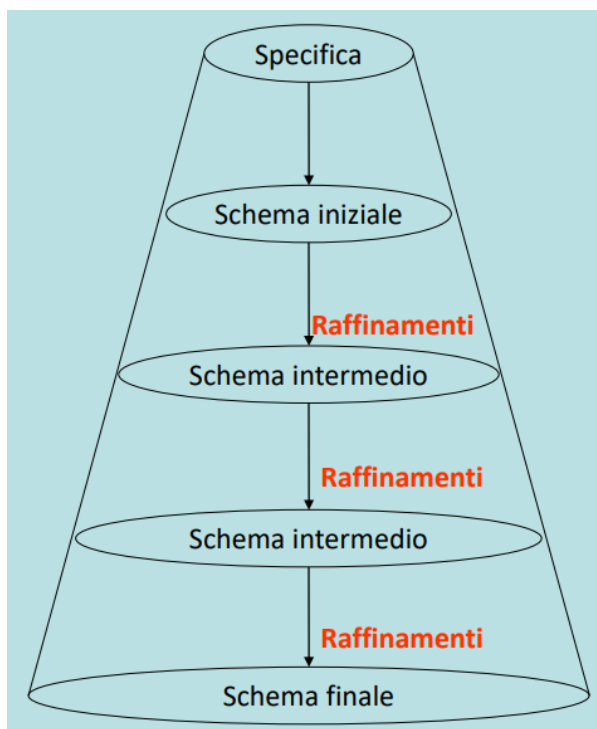
Ripassando i passi da seguire si ha:

- analisi requisiti
- analisi per definire il **glossario dei termini** e **raggruppamento** di requisiti sulla base di termini
- usare glossario con **SINONIMI** per **uniformare** la definizione dei **termini**
- progettazione **concettuale**
- **schema scheletro** (*grossolano*)

Come si ottiene la progettazione concettuale? Dipende dalla grandezza del team e grandezza del progetto.

STRATEGIE DI PROGETTAZIONE CONCETTUALE

- **TOP-DOWN**, *consigliata per il progetto dell'esame*
- **BOTTOM-UP**;
- **GENERALE**



Si producono **via via schemi intermedi più completi** consultando sempre e continuamente i requisiti.

Top-Down

- Le primitive di trasformazione top-down sono regole che operano su un singolo concetto dello schema e lo trasformano in una struttura più complessa che **descrive il concetto con maggiore dettaglio**:

- Trasformazione**

- **T1**: si applica quando un'entità descrive *due concetti diversi* legati fra di loro.



- **T2**: Un'entità è composta da **sotto-entità distinte**.

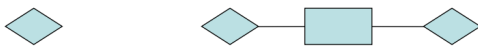


- Trasformazione**

- **T3**: Una relazione in realtà descrive due relazioni diverse tra le stesse entità.



- **T4**: Una **relazione** descrive un concetto con **esistenza autonoma**. In questo caso essa va sostituita con un'entità.



- Trasformazione**

- **T5**: Si applica per aggiungere **attributi ad entità**.

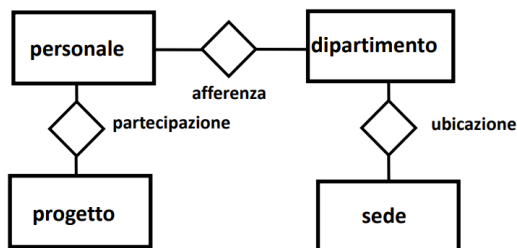


- **T6**: Si applica per aggiungere **attributi a relazioni**.

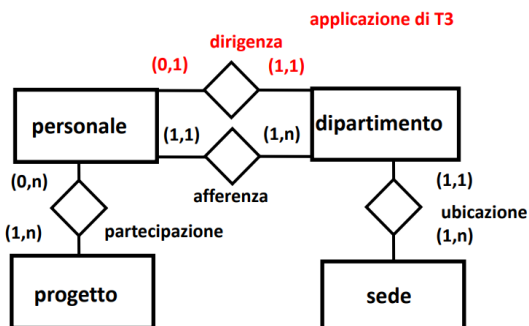


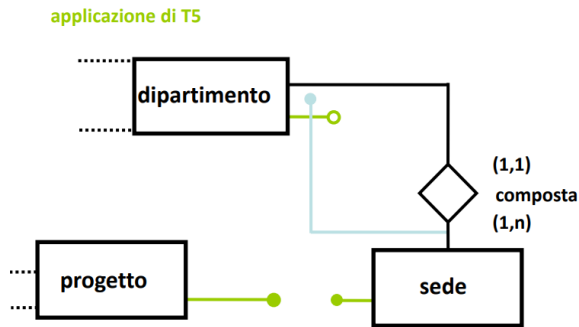
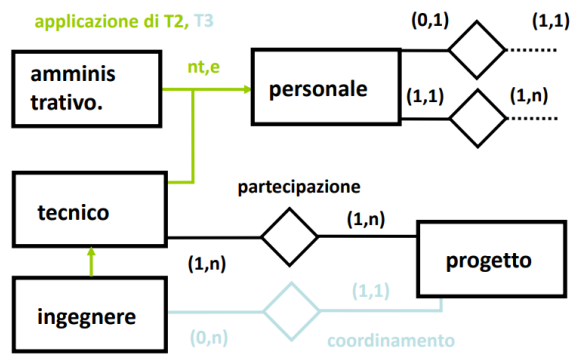
Esempio

schema iniziale



passo 2

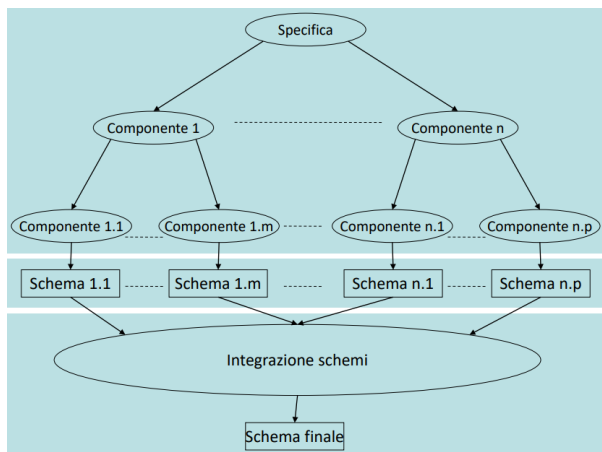




- vantaggi:
 - il progettista descrive inizialmente lo schema trascurando i dettagli
 - precisa lo schema gradualmente
- problema:
 - non va bene per applicazioni complesse perché è difficile avere una visione globale precisa iniziale di tutte le componenti del sistema

Bottom-Up

- Le **specifiche** nascono **suddivise** per **sottoprogetti** descriventi frammenti limitati della realtà da schematizzare;
- si sviluppano i sottoschemi separati
- si **fondono** i **sottoschemi** per ottenere lo schema finale



Primitive di trasformazione Bottom-Up

• Trasformazione

- **T1:** si individua nella specifica una classe di oggetti con proprietà comuni e si introduce un' **entità** corrispondente.

concetto



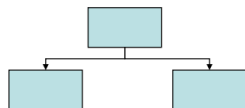
• Trasformazione

- **T2:** si individua nella specifica un legame logico fra entità e si introduce una **associazione** fra esse.



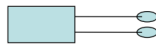
• Trasformazione

- **T3:** si individua una **generalizzazione** fra entità.



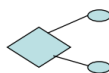
• Trasformazione

- **T4:** a partire da una serie di **attributi** si individua un'entità che li aggrega.

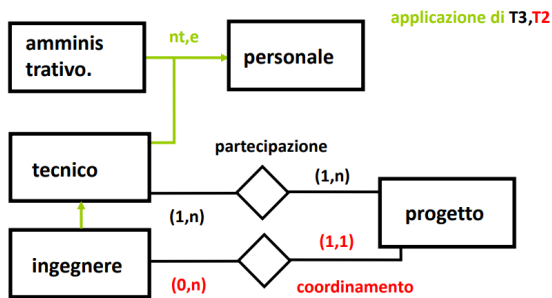


• Trasformazione

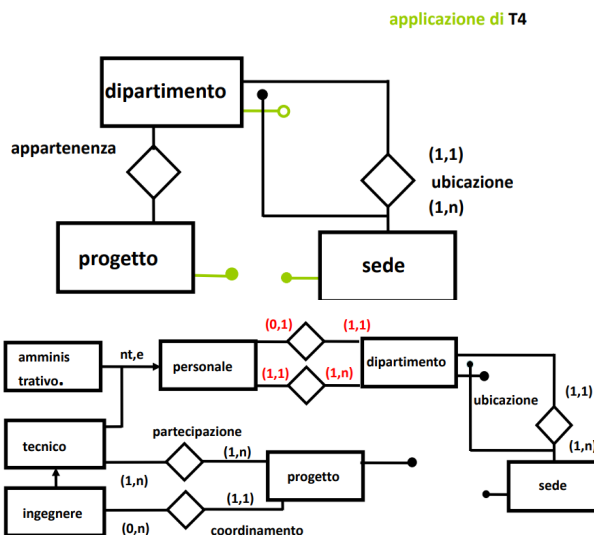
- **T5:** a partire da una serie di **attributi** si individua una relazione che li aggrega.



Sviluppo bottom-up: schema 1



Sviluppo bottom-up: schema 2



Vantaggi e Svantaggi della Strategia Bottom-Up

- Si adatta bene ad una progettazione di gruppo in cui , diversi progettisti possono sviluppare parti disgiunte che possono essere assemblate successivamente.
- L'integrazione di sistemi concettualmente diversi comporta notevoli difficoltà.

Altre strategie

- **inside-out:** è una variante della bottom-up, si sviluppano schemi parziali in aggiunta a sottoschemi già definiti precedentemente e separatamente. Inizialmente si sviluppano alcuni concetti e poi si estendono a macchia d'olio.
- **strategia mista:** cerca di combinare i vantaggi top-down e bottom-up: il progettista divide i requisiti in componenti separate (come nel bottom-up) ma, allo stesso tempo, definisce uno schema scheletro, contenente, a livello astratto, i concetti principali dell'applicazione. Questo fornisce una visione unitaria, anche se astratta, dell'intero progetto e può guidare le fasi di integrazione dei sottoschemi

Metodologia Generale

- **Analisi requisiti**
 - Costruire **glossario** dei termini
 - Analizzare i **requisiti** ed eliminare **ambiguità**
 - Raggruppare i requisiti in insiemi omogenei
- **PASSO BASE**
 - Individuare i concetti più rilevanti e rappresentarli in uno schema scheletro
- **PASSO DI DECOMPOSIZIONE:** *da effettuare se opportuno o necessario*
 - Effettuare una decomposizione dei requisiti con riferimento ai concetti presenti nello schema scheletro
- **PASSO ITERATIVO:** *da ripetere (ALMENO UNA VOLTA) a tutti i sottoschemi (se presenti) finché ogni specifica è stata rappresentata*
 - Raffinare i concetti presenti sulla base delle loro specifiche
 - Aggiungere nuovi concetti allo schema per descrivere specifiche non ancora descritte
- **PASSO DI INTEGRAZIONE:** *da effettuare se sono presenti diversi sottoschemi*
 - Integrare i vari sottoschemi in uno schema generale facendo riferimento allo schema scheletro
- **Analisi di qualità** (fatto **ALMENO UNA VOLTA**)

Se alla fine ho schemi separati, vado a integrarli insieme e poi faccio l'analisi della qualità dello schema.

Qualità di uno Schema Concettuale

- Viene giudicata in base a delle proprietà che lo schema deve possedere:
 - Correttezza
 - Completezza
 - Leggibilità
 - Minimalità

Correttezza e Completezza

- **Correttezza:** se si utilizzano propriamente i costrutti.
 - Gli errori possono essere **SINTATTICI** : uso non ammesso dei costrutti (ad esempio **generalizzazione** fra relazioni)

- o **SEMANTICI** : uso che non rispetta il loro significato (si usa una relazione per descrivere che un'entità è generalizzazione di un'altra).
- **Completezza**: tutti i dati di interesse sono rappresentati nello schema e tutte le operazioni possono essere eseguite a partire dai concetti dello schema
- **Leggibilità**: Uno schema è leggibile quando rappresenta i requisiti in maniera naturale e facilmente comprensibile. Alcune regole:
 - disporre al centro i costrutti con più legami
 - usare linee perpendicolari cercando di minimizzare le intersezioni.
 - Disporre i padri di generalizzazioni sopra i figli
- **Minimalità**: Uno schema è minimale quando **tutte le specifiche sono rappresentate una sola volta**. Non devono contenere **ridondanze** ovvero concetti deducibili da altri oppure cicli di relazioni e generalizzazioni.
 - *Una ridondanza a volte può nascere da una scelta precisa di progettazione*

FASI DELLA PROGETTAZIONE LOGICA

La progettazione logica viene considerata come processo diviso in 2 parti:

1. Ristrutturazione dello schema E-R:

- è una fase indipendente dal modello logico e si basa su criteri di **ottimizzazione dei costi computazionali** dello schema e di successiva semplificazione.

2. Traduzione verso il Modello Logico:

- fa riferimento ad un *modello logico* (ad es. *relazionale*) e può includere ulteriore ottimizzazione che si basa sul modello logico stesso (es. *normalizzazione*).

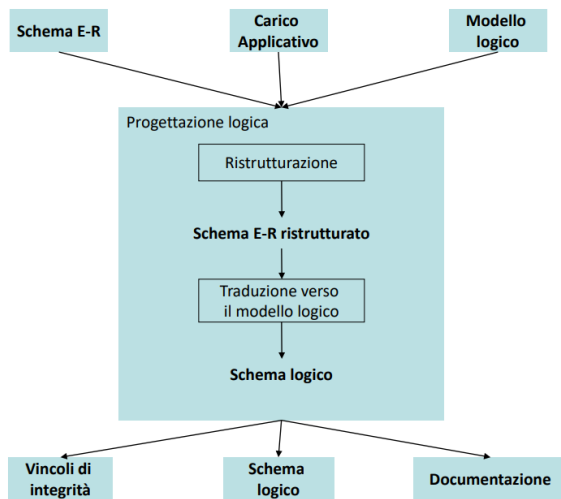
Input ed output della prima fase

Input :

- Schema Concettuale E-R iniziale, Carico Applicativo previsto (in termini di dimensione dei dati e caratteristica delle operazioni)

Output :

- Schema E-R ristrutturato che rappresenta i dati e tiene conto degli aspetti realizzativi (operazioni dopo essere state ottimizzate)



Analisi delle prestazioni su schemi E-R

Gli indici di prestazione per la valutazione di schemi E-R sono due:

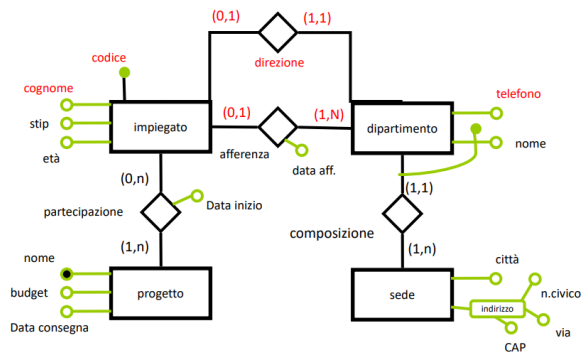
- **Costo di un'operazione:** in termini di numero di occorrenze di entità ed associazioni che mediamente vanno visitate per rispondere a quella operazione sulla base di dati (talvolta sarà necessario raffinare questo criterio)
 - Questo indice risponde alla domanda: *Quanti record devo recuperare per poter rispondere ad una data operazione?*
- **Occupazione di memoria:** viene valutata in termini dello spazio di memoria (misurato in byte) necessario per memorizzare i dati del sistema.

Per studiare questi due parametri abbiamo bisogno di conoscere:

- *Volume dei dati:*
 - a) numero (medio) di occorrenze di ogni entità ed associazione
 - b) dimensioni di ciascun attributo
- *Caratteristiche delle operazioni:*
 - a) tipo di operazione (**INTERATTIVA**[più critica, fatta dall'utente e l'utente vuole una risposta immediata. Sono quelle che vanno ottimizzate come costo] o **BATCH**[eseguite in background, quindi ci interessa di meno])
 - b) frequenza (esecuzioni/tempo[1 volta al mese o più volte all'ora])
 - c) dati coinvolti (entità e o associazioni)

Esempio:

ditta con sedi in città diverse



- **Operazione 1:** assegna un impiegato ad un progetto
- **Operazione 2:** trova i dati di un impiegato, del dipartimento nel quale lavora e dei progetti in cui e' coinvolto
- **Operazione 3:** trova i dati di tutti gli impiegati di un certo dipartimento
- **Operazione 4:** per ogni sede, trova i dipartimenti con il cognome del direttore e l'elenco degli impiegati.

Operazioni sono derivate dai requisiti (dettate dai requisiti). dall'analisi dei requisiti possiamo costruire:

Tabella dei volumi e delle operazioni

TABELLA DEI VOLUMI			TABELLA DELLE OPERAZIONI		
Concetto	Tipo	Volume	Concetto	Tipo	Frequenza
Sede	E	10	Op. 1	I	50 al giorno
Dipartimento	E	80	Op. 2	I	100 al giorno
Impiegato	E	2000	Op. 3	I	10 al giorno
Progetto	E	500	Op. 4	B	2 a settimana
Composizione	R	80			
Afferenza	R	1900			
Direzione	R	80			
Partecipazione	R	6000			

Tipo volume -> Entità oppure Relazione

Tavola dei volumi, quanti record in media sono contenuti all'interno di quella relazione indicata dal "concetto"

es: In Impiegato (entità) si avranno in media 2000 record.

TIPO OPERAZIONE

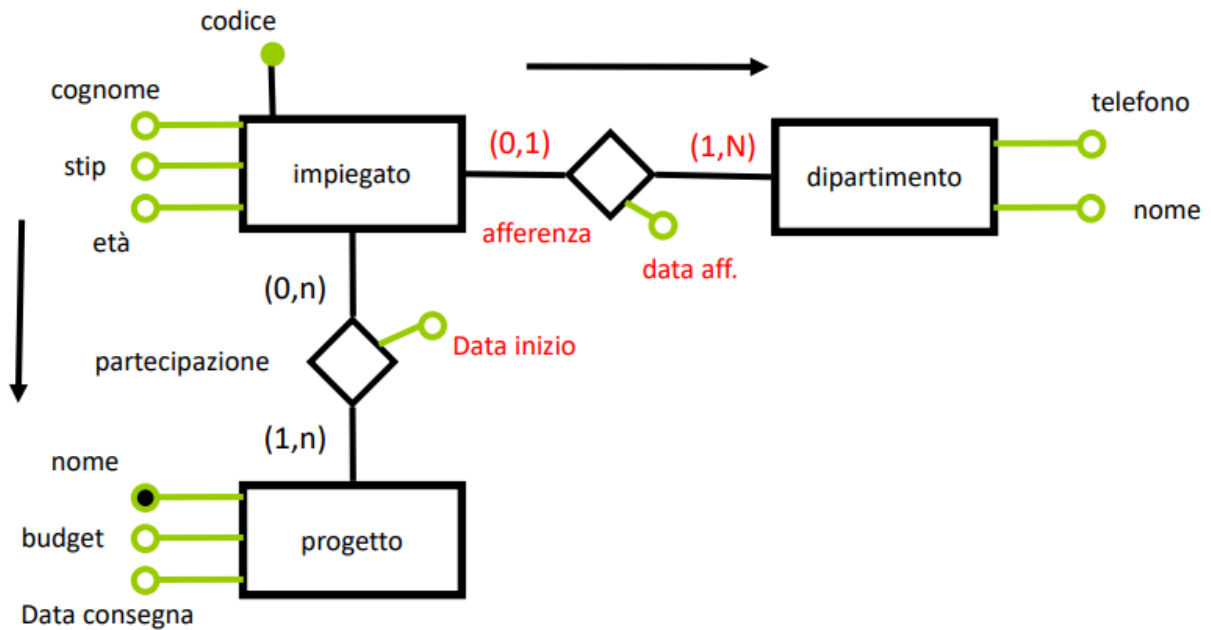
Ci sono 2 tipi di operazioni: Interattiva o Batch

le **batch** sono considerate di meno durante l'ottimizzazione.
quindi sono **meno prioritarie** in analisi dei costi

Avendo a disposizione questi dati è possibile stimare i costi di ogni operazione contando il numero di accessi alle occorrenze di entità e relazioni necessario per eseguire l'operazione.

Prendiamo per esempio *Operazione 2*: trova i dati di un impiegato, del dipartimento nel quale lavora e dei progetti in cui è coinvolto e facciamo riferimento allo schema di operazione. Si assuma che ogni impiegato partecipa in media a 3 progetti. (6000/2000 dalla tab dei volumi)

per ogni operazione si costruisce lo schema dell'operazione che è un'estratto dello schema E-R dove vengono indicati i flussi dei dati con le frecce



Stima del costo dell'operazione 2

- Dobbiamo accedere ad:
 - un'occorrenza di *Impiegato* e di *Afferenza* e quindi di *Dipartimento*;
 - Successivamente, per avere i dati dei progetti a cui lavora, dobbiamo accedere (in media) a tre occorrenze di *Partecipazione* e quindi a tre entità *Progetto*.
 - Tutto viene riassunto nella tavola degli accessi

Tavola degli accessi

TAVOLA DEGLI ACCESSI

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Impiegato	Entita'	1	L
Afferenza	Relazione	1	L
Dipartimento	Entita'	1	L
Partecipazione	Relazione	3	L
Progetto	Entita'	3	L

L lettura, S scrittura. In genere la scrittura e' piu' onerosa della lettura ($1S = 2L$)

La scrittura è più costosa

Tutto questo serve per la ricostruzione di E-R che si divide in 4

Ristrutturazione di schemi E-R

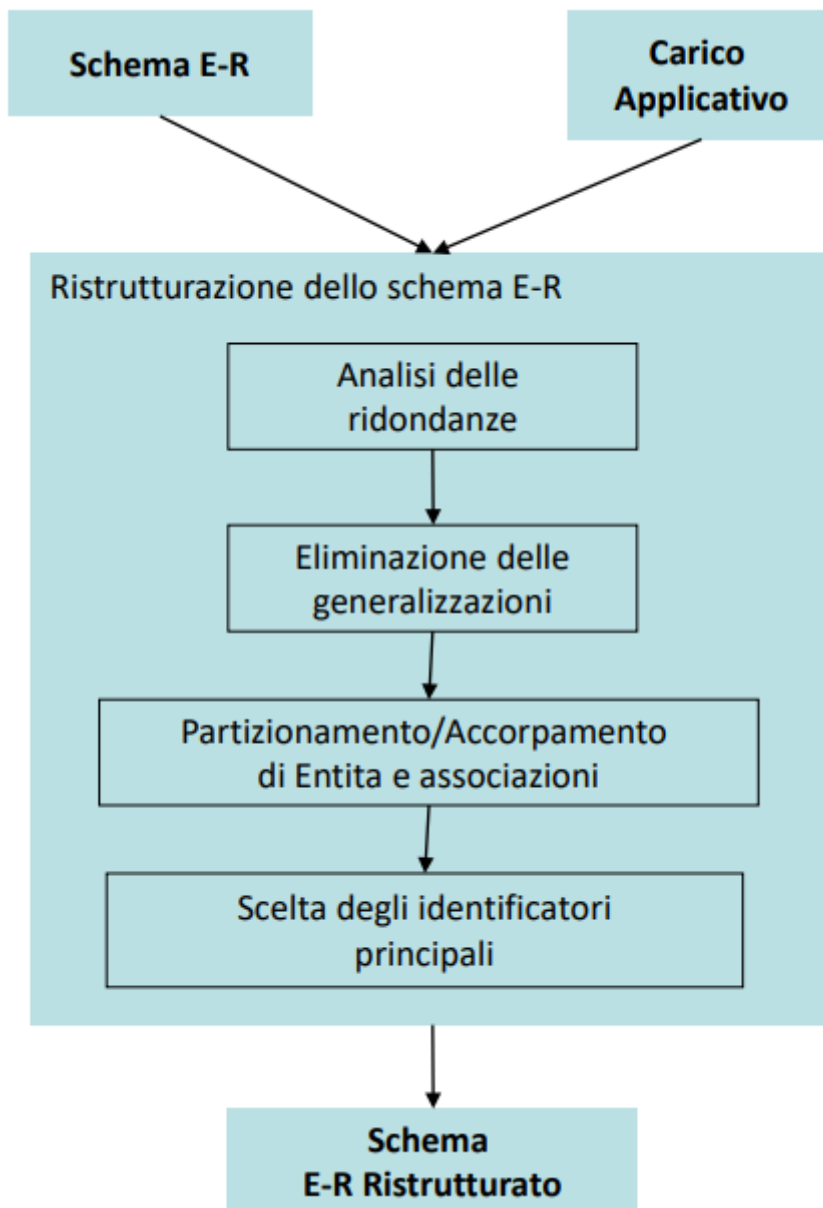
- **Analisi delle Ridondanze:** si decide se eliminare o no eventuali ridondanze.
- **Eliminazione delle Generalizzazioni:** tutte le generalizzazioni vengono analizzate e sostituite da altro.
- **Partizionamento/Accorpamento di entita' ed associazioni:** si decide se partizionare concetti in piu' parti o viceversa accorpare.
- **Scelta degli identificatori primari:** si sceglie un identificatore per quelle entita' che ne hanno piu' di uno

RIDONDANZA vuol dire usare attributi in più che si possono calcolare anche da altre entità.
Eliminazione delle Generalizzazioni: tutte le generalizzazioni vengono analizzate e sostituite da altro

- ristrutturazione delle generalizzazione in base ai costi.

Partizionamento/Accorpamento di entità ed associazioni: si decide se partizionare concetti in più parti o viceversa accorpare.

Scelta degli identificatori primari: si sceglie un identificatore per quelle entità che ne hanno più di uno. Si vanno a valutare con i costi.



ANALISI DELLE RIDONDANZE

- Attributi derivabili da altri attributi della stessa entità (fattura: importo lordo)
- Attributi derivabili da attributi di altre entità (o associazioni) (Acquisto: Importo totale da Prezzo)
- Attributi derivabili da operazioni di conteggio (Città: Numero abitanti contando il numero di Residenza)
- Associazioni derivabili dalla composizione di altre associazioni in presenza di cicli. Tuttavia i cicli non necessariamente generano ridondanze

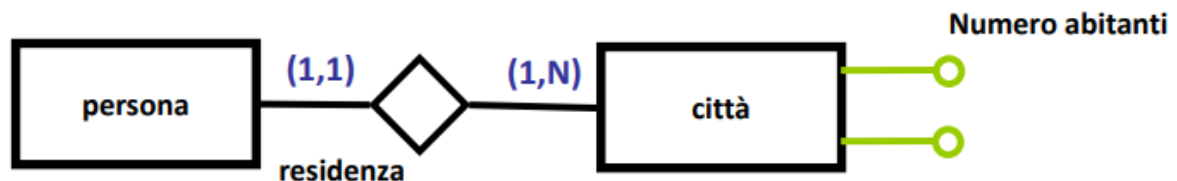
quando si ha un dato ridondante derivabile, esso:

- **Vantaggi:** riduce gli accessi per calcolare il dato derivato.
- **Svantaggi:** occupazione di memoria e necessità di effettuare operazioni aggiuntive per mantenere il dato aggiornato.

Decisione: mantenere o eliminare?

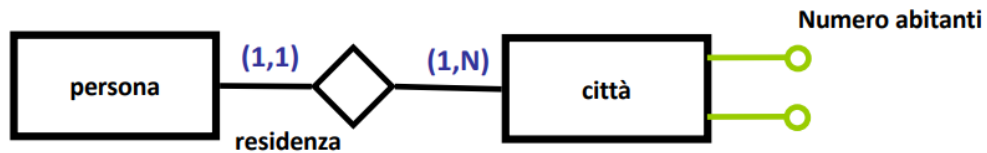
- **Basta confrontare i costi di esecuzione delle operazioni sull'oggetto con e senza ridondanza**

Esempio di ridondanza:



numero abitanti è una ridondanza perchè tale attributo deriva da un conteggio in "persona" ma, analizzando i costi, conviene mantenerla:

- **Consideriamo l'esempio Città-Persona per l'anagrafica di una regione.**
 - *Operazione 1:* memorizza una persona nuova con la relativa città.
 - *Operazione 2:* stampa tutti i dati di una città (incluso il numero di abitanti).
- **Valutiamo gli indici di prestazione per l'attributo Numero Abitanti**



Concetto	Tipo	Volume
Città	E	200
Persona	E	1000000
Residenza	R	1000000

Operazione	Tipo	Frequenza
Op. 1	I	500 al giorno
Op. 2	I	2 al giorno

Valutazione in presenza della ridondanza

Costo memoria per l'attributo NUMABITANTI:

Assumendo che il numero di abitanti richieda 4 byte il dato richiede $4 \times 200 = 800$ byte.

- Operazione 1 richiede
 - un accesso in scrittura a Persona
 - uno in scrittura a Residenza
 - uno in lettura per cercare la città'
 - ed uno in scrittura (per incrementare il numero di abitanti) a Città
 - ripetuto 500 volte
 - » si hanno 1500 accessi in scrittura e 500 in lettura.
- L'operazione 2 richiede
 - un solo accesso in lettura a Città
 - 2 volte al giorno. (trascurabile...)
- Supponendo che la scrittura ha un costo doppio rispetto ad una lettura si hanno 3500 accessi al giorno in presenza della ridondanza.

Valutazione in assenza della ridondanza

- Per l'operazione 1,
 - un accesso in scrittura a Persona
 - ed uno in scrittura a Residenza
 - Ripetuto 500 volte
 - un totale di 1000 accessi in scrittura al giorno.
- Per l'operazione 2 abbiamo bisogno di un accesso
 - in lettura a Città (possiamo trascurare) e
 - 5000 accessi in lettura a Residenza in media (persone/città)
 - Ripetuto 2 volte al giorno
 - per un totale di 10.000 accessi in lettura al giorno.
- Considerando doppi gli accessi il scrittura, il totale e' di 12000. Quindi 8500 in più rispetto al caso di ridondanza contro meno di un solo Kilobyte di memoria in più.

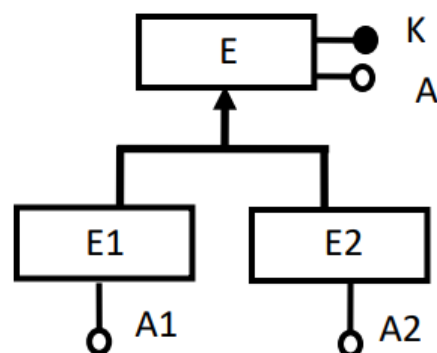
ANALISI DELLE GERARCHIE

Per ogni ridondanza si va a valutare il costo con ridondanza oppure senza e si valuta sulla base del costo minore

Eliminazione delle gerarchie

il modello relazionale non rappresenta le gerarchie, le gerarchie sono sostituite da entità e associazioni:

- 1) mantenimento delle entità con associazioni
- 2) collasso verso l'alto
- 3) collasso verso il basso



l'applicabilità e la convenienza delle soluzioni dipendono dalle proprietà di copertura e dalle operazioni previste

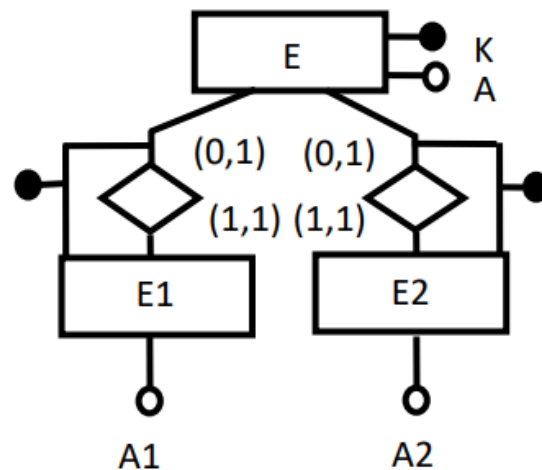
Non sempre le 3 strategie sono applicabili ma i costi, spesso, guidano la scelta.
La scelta delle 3 che garantisce **minor costo**, è quella che andrò ad effettuare.

Mantenimento delle entità

Mantengo tutte le entità E1 e E2 ma sostituisco la generalizzazione con 2 relazioni di tipo 1:1 tipicamente

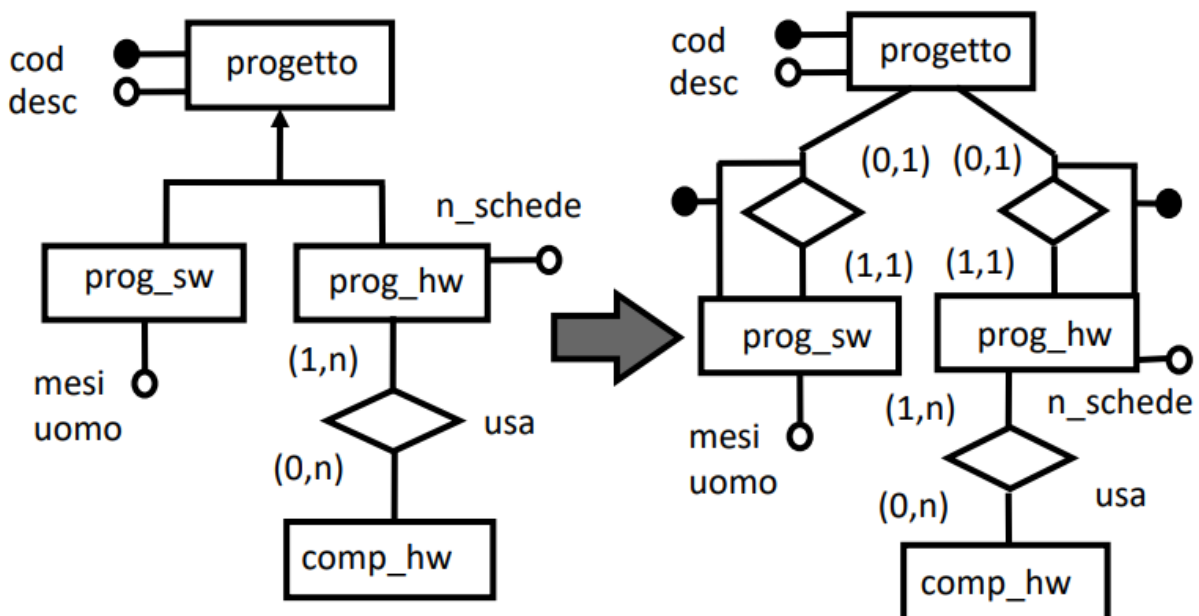
mantenimento delle entità

- tutte le entità vengono mantenute
- le entità figlie sono in associazione con l'entità padre
- le entità figlie sono identificate esternamente tramite l'associazione



questa soluzione è sempre possibile,
indipendentemente dalla copertura

mantenimento entità - es.:

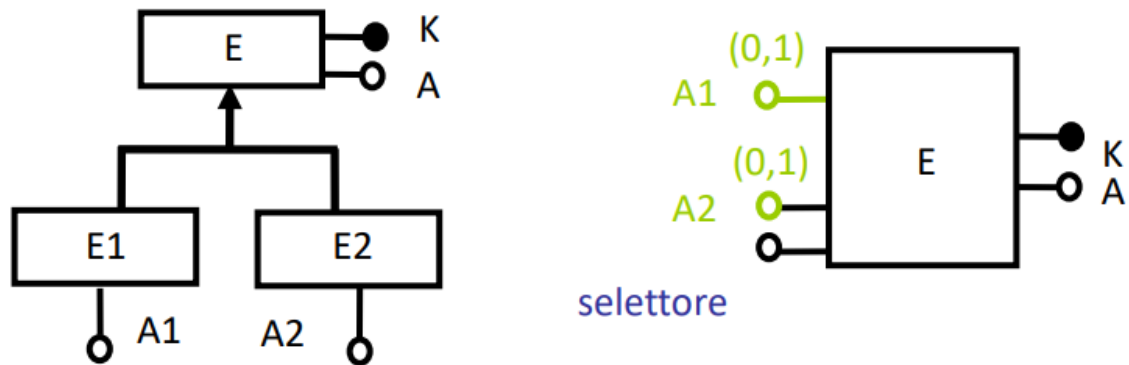


Collasso verso l'alto:

Le due specializzazioni $E1$ e $E2$ vengono unite al padre E

eliminazione delle gerarchie

- Il **collasso verso l'alto** riunisce tutte le entità figlie nell'entità padre



selettore è un attributo che specifica se una istanza di E appartiene a una delle sottoentità

Il collasso verso l'alto favorisce operazioni che consultano insieme gli attributi dell'entità padre e quelli di una entità figlia: – in questo caso si accede a una sola entità, anziché a due attraverso una associazione

- gli attributi obbligatori per le entità figlie divengono opzionali per il padre
- si avrà una certa percentuale di valori nulli

Il collasso verso l'alto non è sempre applicabile.

" E " è una gerarchia non esclusiva (*ne*), allora posso avere entità che può essere $E2$ oppure $E1$ e quindi il selettore da solo non è sufficiente.

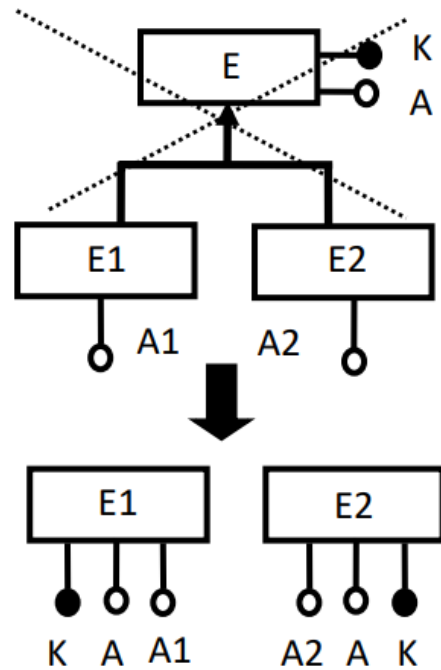
Collasso verso il basso

limiti di applicabilità:

- se la copertura non è totale
non si può fare:

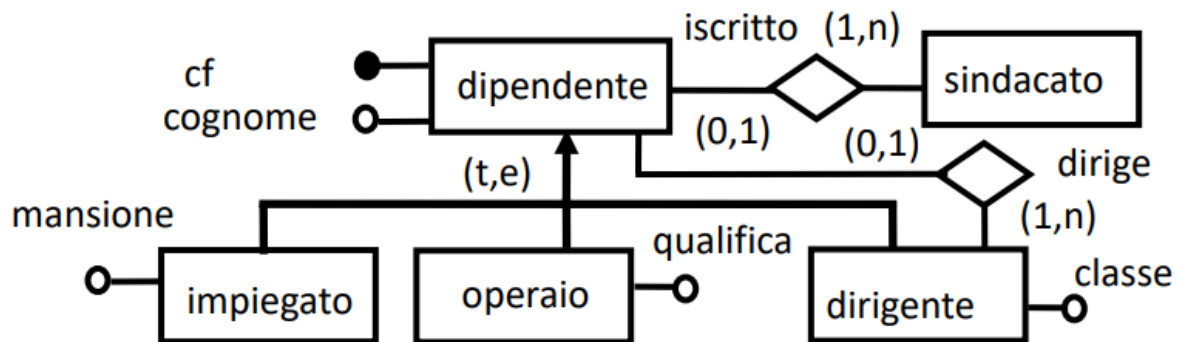
dove mettere gli E che non sono né E1, né E2 ?

- se la copertura non è esclusiva introduce
ridondanza: per una istanza presente sia in E1 che in E2 si rappresentano due volte gli attributi di E



Se ho più strade praticabili, allora valuto il costo delle operazioni in base alle scelte che sto facendo e in base ai costi decido come eliminare le gerarchie.

collasso verso il basso: es.



collasso verso il basso: es.

