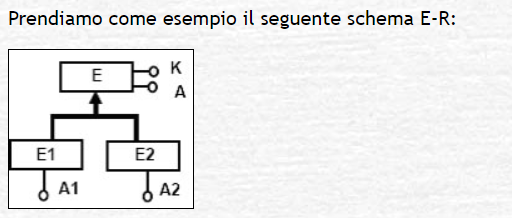
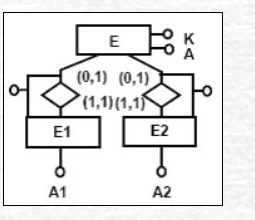
Riferimento: <http://www.dacrema.com/Informatica/gerarchie_elimin.htm>

Per la generalizzazione esistono tre modi di eliminare le gerarchie:

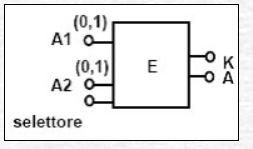


1) Mantenimento delle entità con associazioni



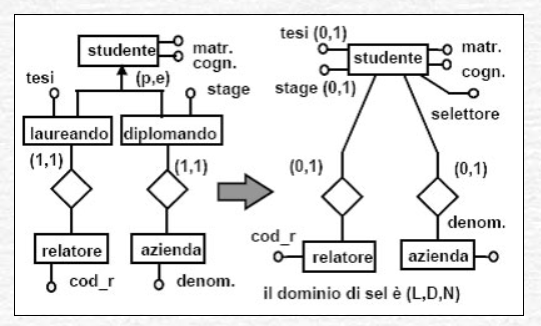
Tutte le entità vengono mantenute. Le entità figlie vengono messe in associazione con l’entità padre e sono identificate esternamente tramite l’associazione. La cardinalità (0,1) indica che per tale associazione l’ entità padre può avere zero o una entità figlio. Mentre (1,1) che l’ entità figlio può avere un solo padre.

2) Collasso verso l’alto

Esso riunisce tutte le entità figlie nell’entità padre.

E1 ed E2 vengono eliminate e le loro proprietà vengono aggiunte all’ entità padre . Gli attributi obbligatori per le entità figlie divengono opzionali per il padre , indicato dalla cardinalità (0,1), con la conseguenza che si *avrà una certa percentuale di valori nulli*. All’ entità ottenuta viene aggiunto un ulteriore attributo (selettore) che specifica se una istanza di E appartiene a una delle sottoentità.

Se la gerarchia era totale ed esclusiva il selettore ha N valori, quante sono le sottoentità. Se la gerarchia era parziale esclusiva il selettore ha N+1 valori; il valore in più serve per le istanze che non appartengono ad alcuna sottoentità. Se avessimo un overlapping occorrerebbero tanti selettori booleani quante sono le sottoentità. Vediamo di seguito un esempio di uno schema in cui è stata collassata la gerarchia verso l’ alto:



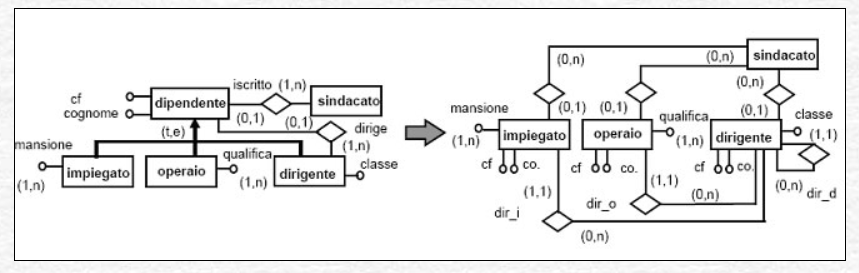
3) Collasso verso il basso

Si elimina l’entità padre trasferendone gli attributi su tutte le entità figlie. Una associazione del padre è replicata, tante volte quante sono le entità figlie la soluzione è interessante in presenza di molti attributi di specializzazione (con il collasso verso l’alto si avrebbe un eccesso di valori nulli).

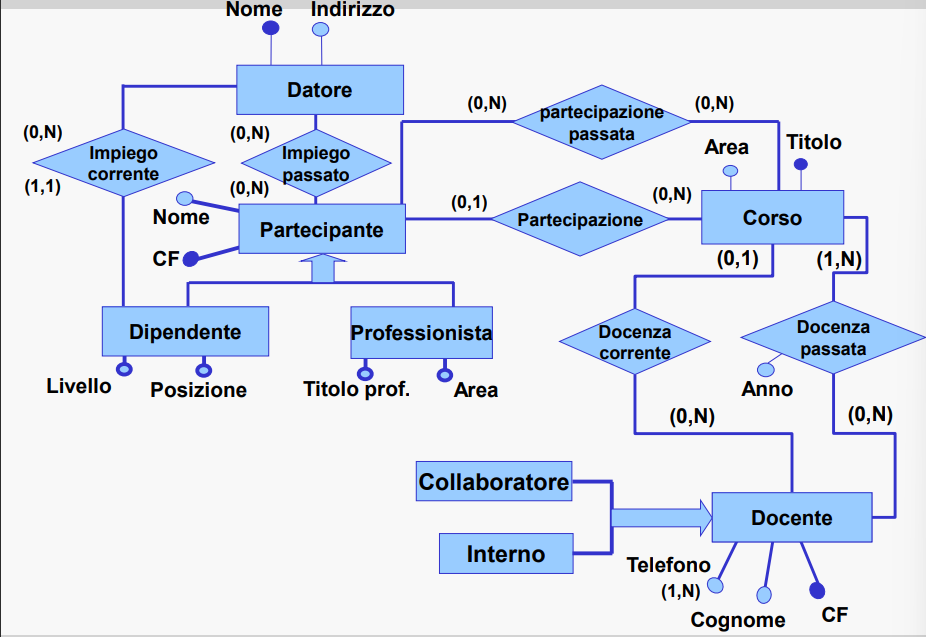
Immagine che contiene testo, orologio

Descrizione generata automaticamente

Si noti che se la copertura è parziale non si può fare il collasso verso il basso. Dove mettere gli E che non sono né E1, né E2 ? Mentre se la copertura è overlapping introduce ridondanza, infatti per una istanza presente sia in E1 che in E2 si rappresentano due volte gli attributi di E.

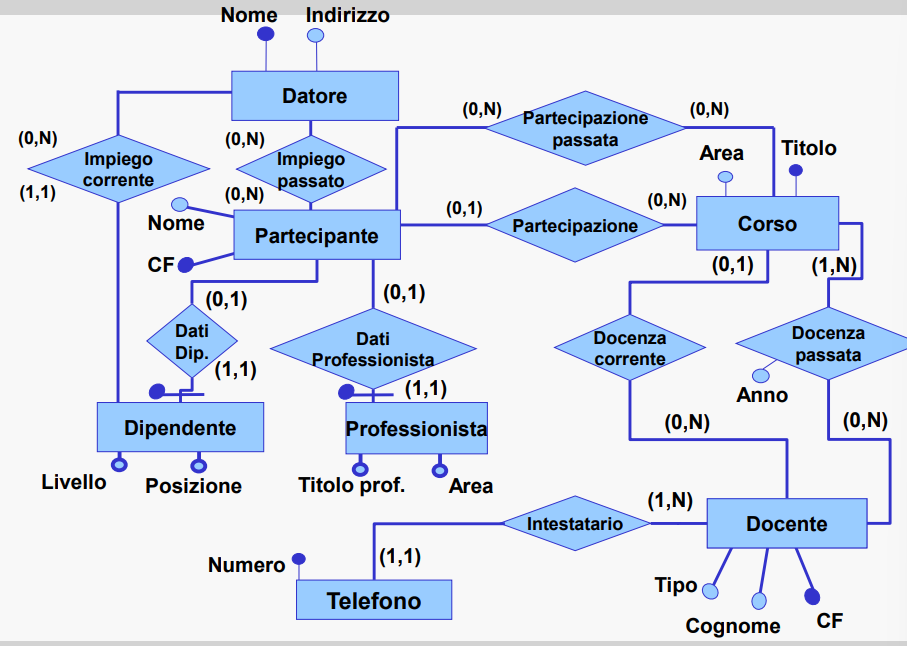
Vediamo di seguito un esempio di uno schema in cui è stata collassata la gerarchia verso il basso:

L’esempio visto in classe parte dallo schema:



E attua una minimizzazione dei valori nulli:

* mantenendo Docenza Corrente e Docenza Passata come relazioni
* usando il mantenimento delle entità con associazioni tra Partecipante e Dipendente/Professionista
* creando per l’attributo multivalore Telefono, un’entità apposita
* mantenendo Partecipazione e Partecipazione Passata come relazioni e similmente anche per Impiego (Corrente e Passato)



Creando come schema

• Partecipante(CF, Nome, Corso-Attuale)

• Partecipante.Corso-Attuale → Corso.Titolo

• Datore(Nome, Indirizzo)

• Corso(Titolo, Area, CF-Docente-Corrente)

• Corso.CF-Docente-Corrente → Docente.CF

• Docente(CF, Cognome,Tipo)

• Docente.Tipo ∈ { Collaboratore, Interno }

• Telefono(Numero, CF-Docente)

• Telefono.CF-Docente → Docente.CF

• Dipendente(CF, Livello, Posizione, Datore)

• Dipendente.CF → Partecipante.CF

• Dipendente.Datore → Datore.Nome

• Professionista(CF,Titolo-Prof, Area)

• Professionista.CF → Partecipante.CF

• ImpiegoPassato(CF-Partecipante,Nome-Datore)

• ImpiegoPassato.CF-Partecipante → Partecipante.CF

• ImpiegoPassato.Nome-Datore → Datore.Nome

• PartecipazionePassata(CF-Partecipante,Titolo-Corso)

• PartecipazionePassata.CF-Partecipante → Partecipante.CF

• PartecipazionePassata.Titolo-Corso → Corso.Titolo

• DocenzaPassata(CF-Docente,Titolo)

• DocenzaPassata.CF-Docente → CF.Docente

• DocenzaPassata.Titolo → Corso.Titolo

Esercizi concreti

Il sistema è condiviso tra diverse istituzioni, di ognuna delle quali è di interesse sapere il nome dell’istituzione, la via e il codice ISTAT del comune in cui ha la sede principale.

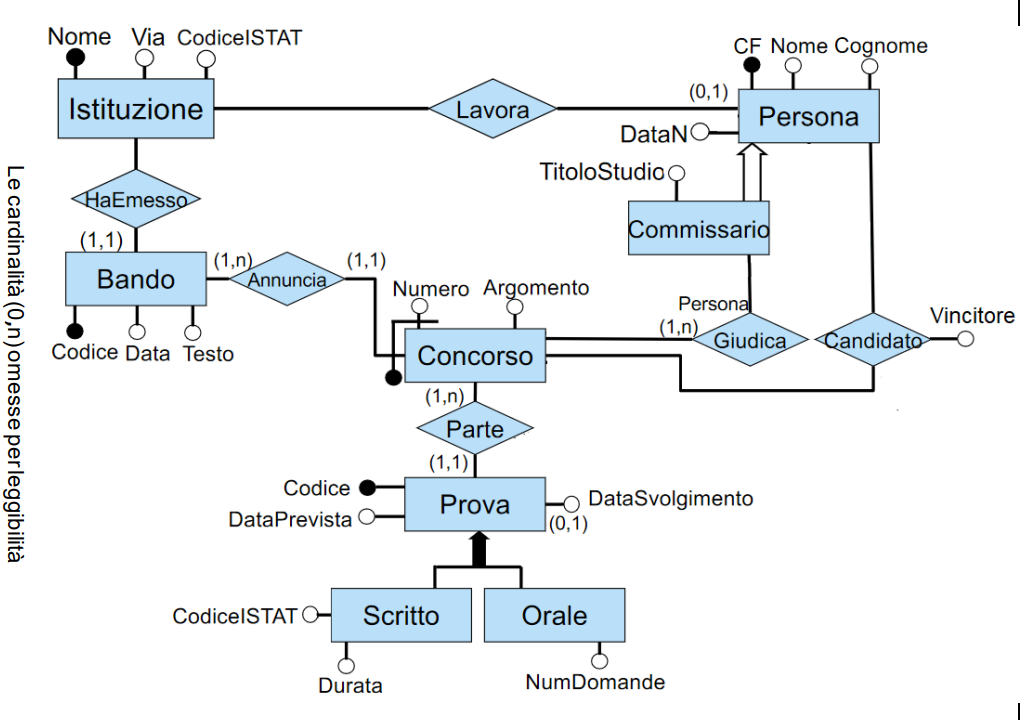
Per ogni istituzione, si vuole sapere le persone che lavorano presto tale istituzione ed i bandi di concorso che ha eventualmente emesso.

Di ogni persona interessa il codice fiscale (identificativo), il nome, il cognome, la data di nascita. Una persona può lavorare presso una istituzione (ma anche essere disoccupata). Alcune persone posso anche fungere da commissari giudicatori di concorso: per loro, si vuole sapere quale sia il titolo di studio.

Di ogni bando interessa il codice identificativo, la data di pubblicazione, il testo ed i concorsi (almeno uno) che sono annunciati nel bando stesso.

Di ogni concorso interessa il bando in cui è annunciato, il numero (unico nell’ambito del bando in cui è annunciato ma diversi bandi possono avere lo stesso numero concorso), l’argomento del concorso (giurisprudenza, ingegneria civile, amministrazione, ecc.) e le prove di cui è composto (almeno una), indentificata da un codice identificativo. Per ogni prova, si vuole conosce la data di svolgimento prevista e successivamente la data di svolgimento effettiva, quando questa avviene.

Le prove sono o scritte o orali e di ogni prova scritta interessa anche la durata e il codice ISTAT del comune dove si prevede si svolga, mentre di ogni prova orale interessa anche il numero di domande previste.

Di ogni concorso interessano anche le persone (almeno una) che hanno partecipato come commissari, le persone che hanno partecipato come candidati (se ci sono) e, tra queste ultime, quelle che sono risultate vincitrici (se ci sono).

*Occorre ottimizzare la seguente operazione: quando si accede alla durata di una prova o al numero di domande, si vuole sempre conoscere il concorso di cui fa parte*.

Quindi si opera eliminando l’entità *Prova* e duplicando i suoi attributi per *Scritto* ed *Orale*.

In merito a Commissario, si mantiene una relazione con cardinalità (0,1) tra Persona e Commissario (perché una persona può essere Commissario) e (1,1) tra Commissario e Persona (perché un commissario è sempre una persona).

Candidato rimane una relazione, essendoci due cardinalità (0,n) e (0,n), aggiungendo solo Vincitore come attributo (può essere nullo in un primo momento come attributo, dato che il testo indica che *può* esserci un vincitore; similmente, essendo che *possono* esserci dei candidati, nella stessa relazione candidato, potrà essere nullo anche l’attributo Persona)

Essendo inoltre che una Persona può essere disoccupata, può essere nullo il collegamento di chiave esterna con Istituzione.

In nome di tutto ciò, si struttura così lo schema logico (indicando con \* i valori nulli)

Istituzione (Nome, CodiceISTAT, Via)

Persona (CF, Nome, Cognome, Data\_nascita, Istituzione\*)

FK:

Persona.Istituzione 🡪 Istituzione.Nome

Commissario (CF, Titolo\_studio, Persona)

FK:

Commissario.CF 🡪 Persona.CF

Giudice (Commissario, Concorso)

FK:

Giudice.Commissario 🡪 Commissario.CF

Giudce.Concorso 🡪 Concorso.Numero

Candidato (Persona, Vincitore\*, Concorso)

FK

Cnndidato.Persona 🡪 Persona.CF

Candidato.Concorso 🡪 Concorso.Numero

Concorso (Numero, Argomento, Bando)

FK

Concorso.Numero 🡪 Bando.Codice

Bando (Codice, Data\_pubb, Testo, Istituzione)

FK

Bando.Istituzione 🡪 Istituzione.Nome

Prova\_scritta (Codice, DataPrevista, DataSvolgimento, CodiceISTAT, Durata, Concorso)

FK

Prova\_scritta.Concorso 🡪 Concorso.Numero

Prova\_orale (Codice, DataPrevista, DataSvolgimento, NumDomande, Concorso)

FK

Prova\_orale.Concorso 🡪 Concorso.Numero