



Progettazione e sviluppo di applicazioni web

Azienda Speciale di Formazione “Scuola Paolo Borsa”
Ing. Masciadri Andrea <masciadri.andrea@gmail.com>

Design di una base di dati

Progetto top-down e bottom-up

- Top-down: si procede per raffinamenti a partire da una descrizione che comprende TUTTA la realtà di interesse.
- Bottom-up: si disegnano separatamente aspetti della realtà e poi li si integra costruendo un unico schema.

Confronto

- Top down: il progetto è più ordinato e razionale, ma più difficile (il progettista deve possedere una “visione d’assieme”).
- Bottom-up: si possono prendere decisioni differenti nell’affrontare i sotto-problemi, che si tradurranno in “conflitti” (modelli diversi della stessa realtà).

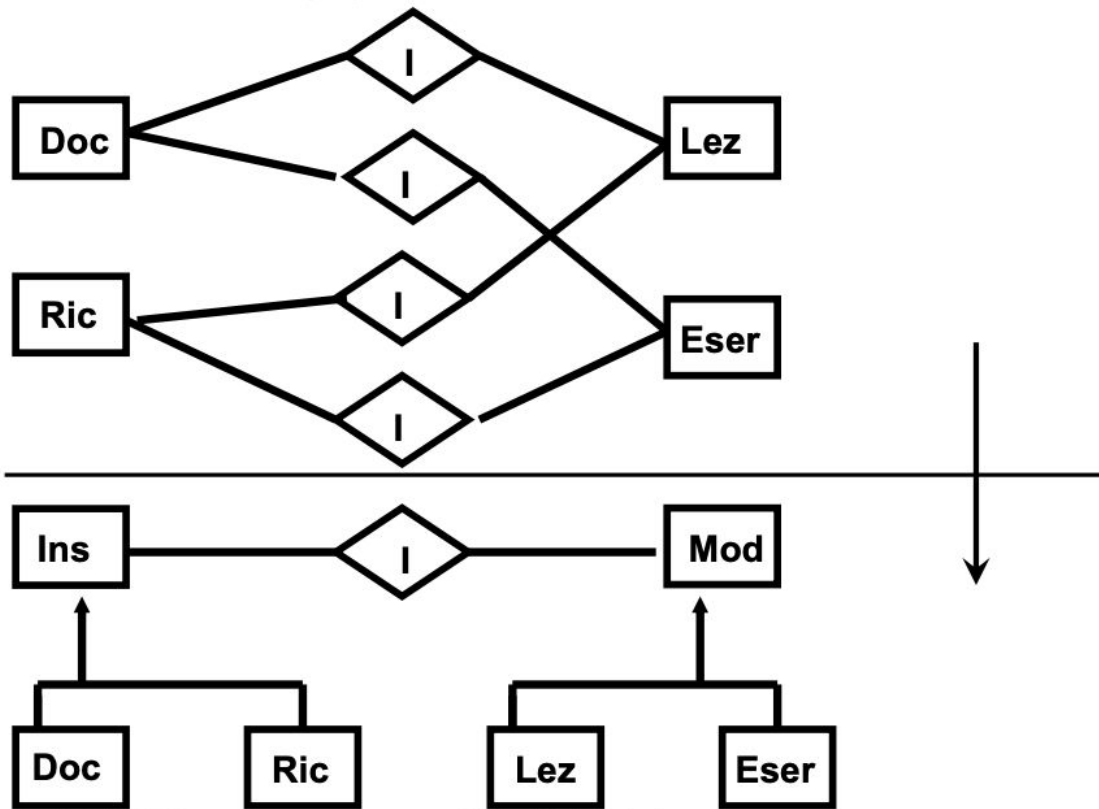
Qualità di uno schema concettuale

- Completezza
- Correttezza
- Leggibilità
- Minimalità
- Auto-esplicatività

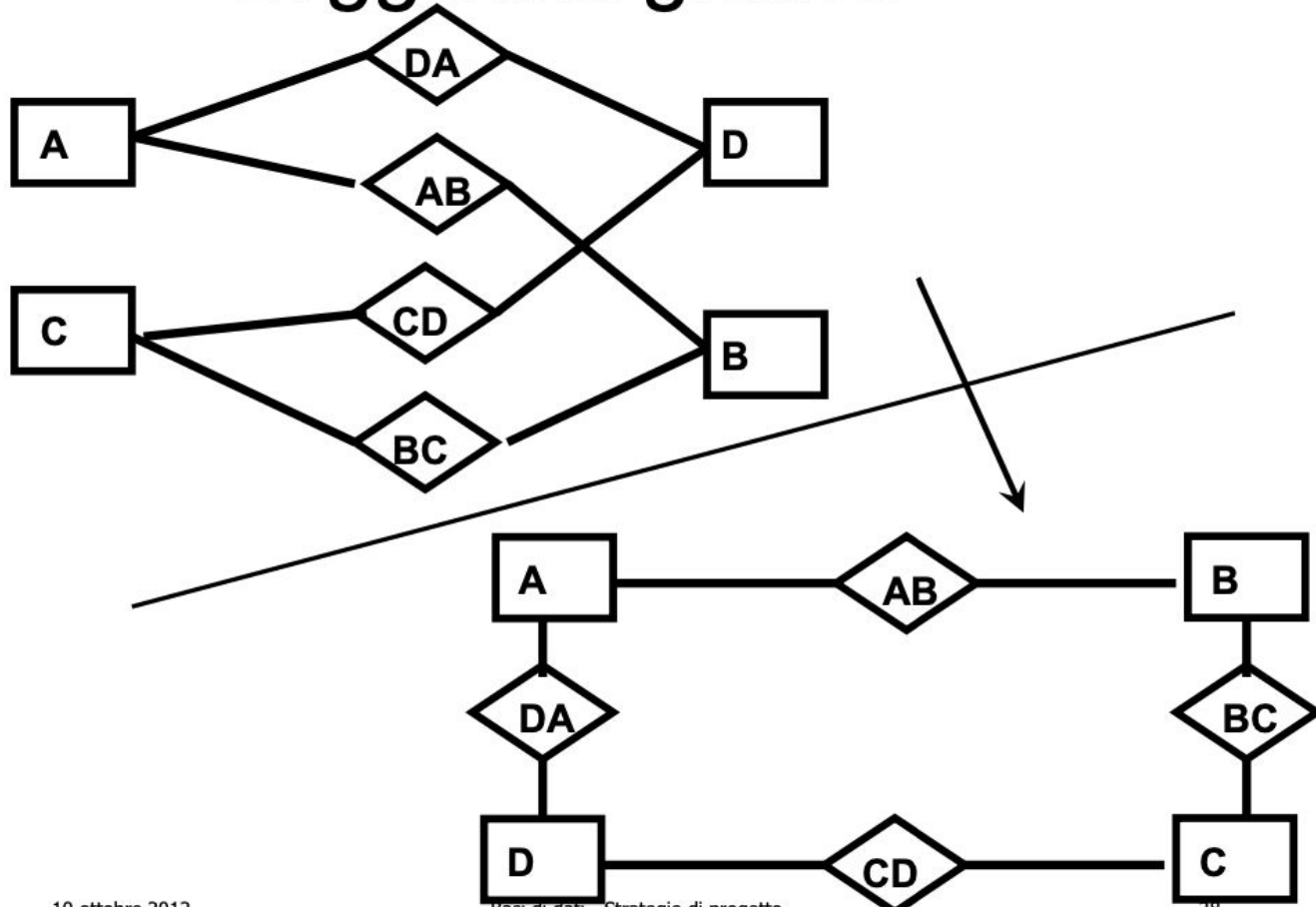
Completezza e correttezza

- Rappresentare in modo completo e corretto i requisiti.
- Sono proprietà ovvie ma sulle quali c'è poco da aggiungere:
 - per la completezza: assicurarsi che i dati consentano di eseguire tutte le applicazioni;
 - per la correttezza: assicurarsi che sia possibile popolare la base di dati anche con informazione parzialmente incompleta durante fasi iniziali della sua evoluzione.

Leggibilità concettuale



Leggibilità grafica



Post-processing

- Verificare che:
 - tutte le entità abbiano un identificatore;
 - tutte le associazioni abbiano cardinalità ben definite;
 - le entità siano significative (consentano di rappresentare più di una attributo o siano collegate ad altre entità tramite associazioni);
 - le generalizzazioni siano utili (consentano di ereditare proprietà).

Progetto logico

Progetto logico

- • Lo schema E/R descrive un dominio applicativo ad un dato livello di astrazione.
- Lo schema E/R serve per:
 - fornire una descrizione sintetica e visiva;
 - rappresentare buona parte della semantica dell'applicazione;
 - scambiare informazioni sull'attività progettuale tra i membri del team di progetto e mantenere una documentazione.

Progetto logico

- Non esistono DBMS in grado di operare direttamente sui concetti di schemi E/R:
 - è quindi necessario tradurli in altri schemi di dati (logico relazionale in queste lezioni);
 - questa traduzione può essere eseguita in modo semi-automatico;
 - le scelte alternative devono tenere conto dell'efficienza dello schema logico risultante e delle operazioni da effettuare (derivanti da flussi e processi).

Fasi del progetto

Il progetto produce trasformazioni e traduzioni dello schema E/R con le seguenti fasi:

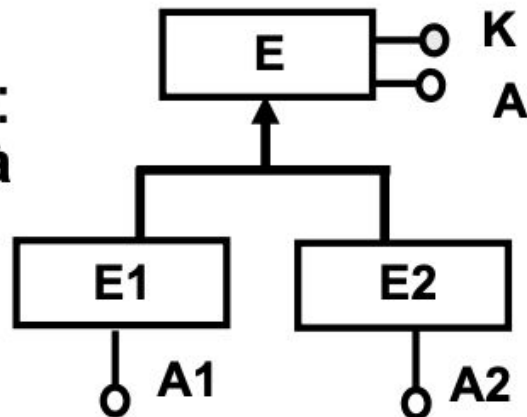
- 1 eliminazione delle gerarchie isa;
- 2 selezione delle chiavi primarie, eliminazione delle identificazioni esterne;
- 3 normalizzazione degli attributi composti o multipli;
- 4 traduzione di entità e associazioni in schemi di relazioni;
- 5 verifica di normalizzazione.

Eliminazione delle gerarchie

Il modello relazionale non rappresenta le gerarchie: le gerarchie sono sostituite da entità e associazioni.

Le possibili soluzioni sono:

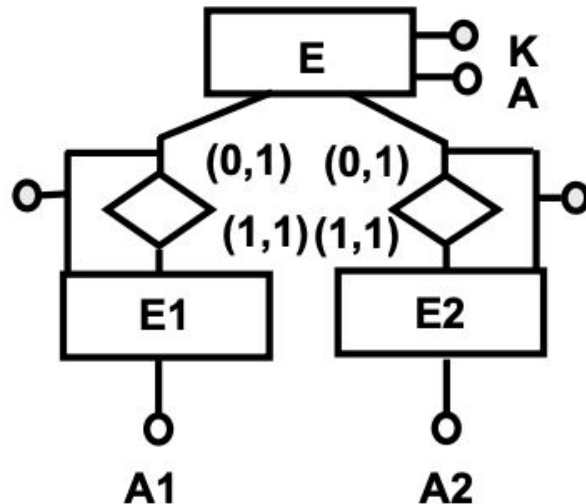
1. **mantenimento delle entità con associazioni**
2. **collasso verso l'alto**
3. **collasso verso il basso**



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

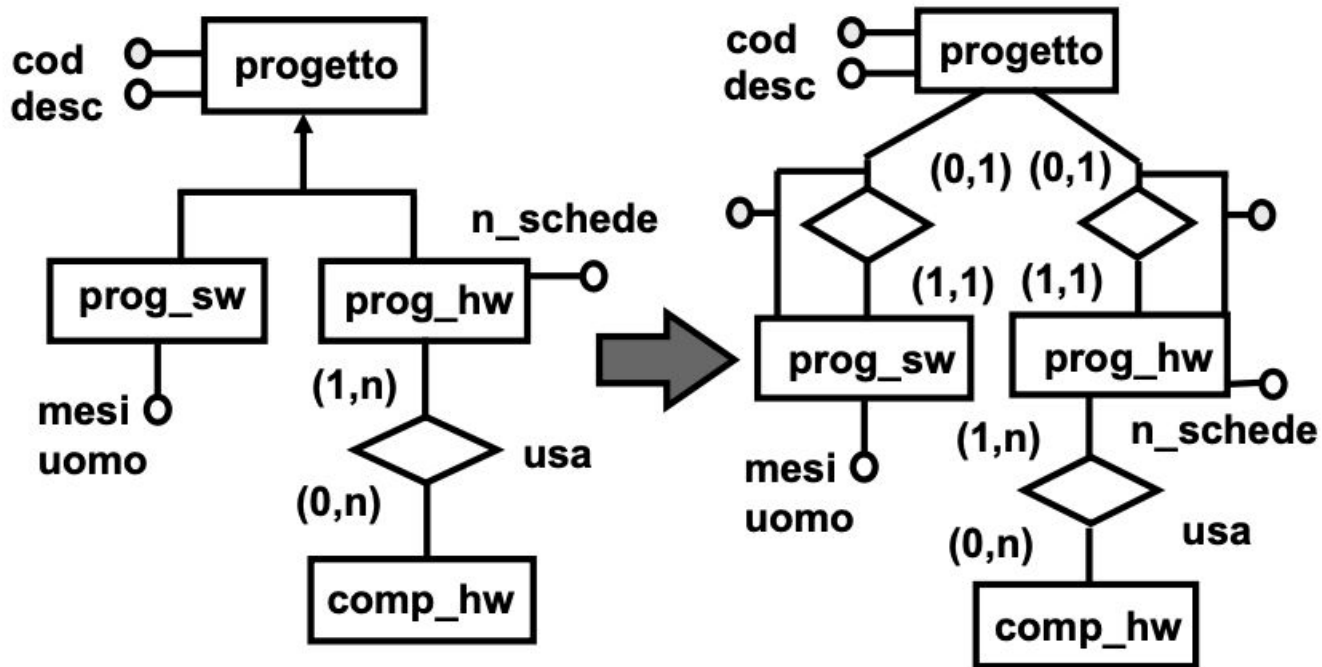
Mantenimento delle entità

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



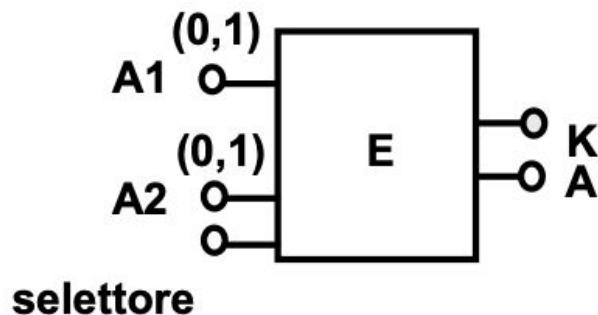
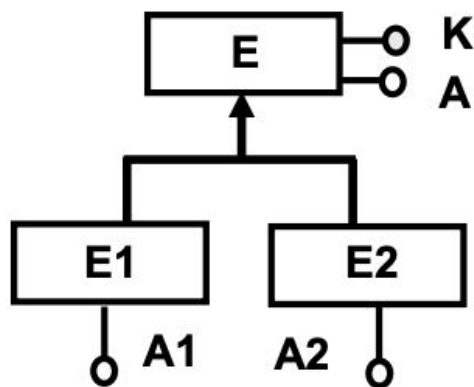
**Questa soluzione è sempre possibile,
indipendentemente dalla copertura.**

Mantenimento entità - esempio



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.

Isa: collasso verso l'alto

- 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.

Isa: collasso verso il basso

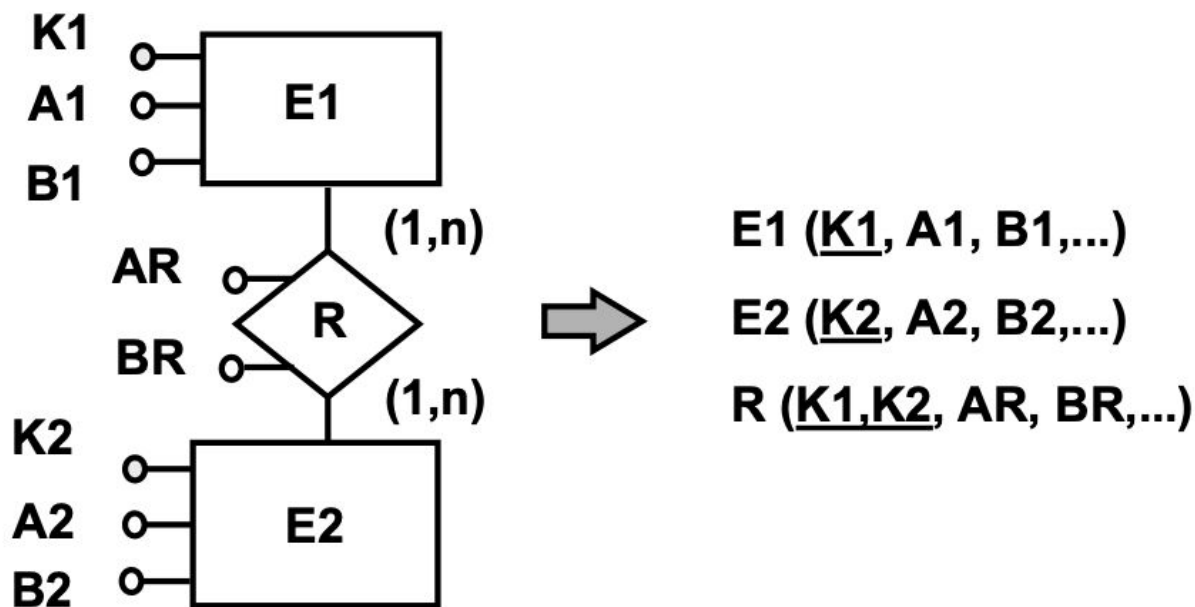
Collasso verso il basso:

- 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);
 - favorisce le operazioni in cui si accede separatamente alle entità figlie.

Traduzione standard

- Ogni entità è tradotta con una relazione con gli stessi attributi:
 - la chiave è l'identificatore dell'entità stessa (già visto).
- Ogni associazione è tradotta con una relazione con gli stessi attributi, cui si aggiungono gli identificatori di tutte le entità che essa collega (già visto):
 - la chiave è composta dalle chiavi delle entità collegate.

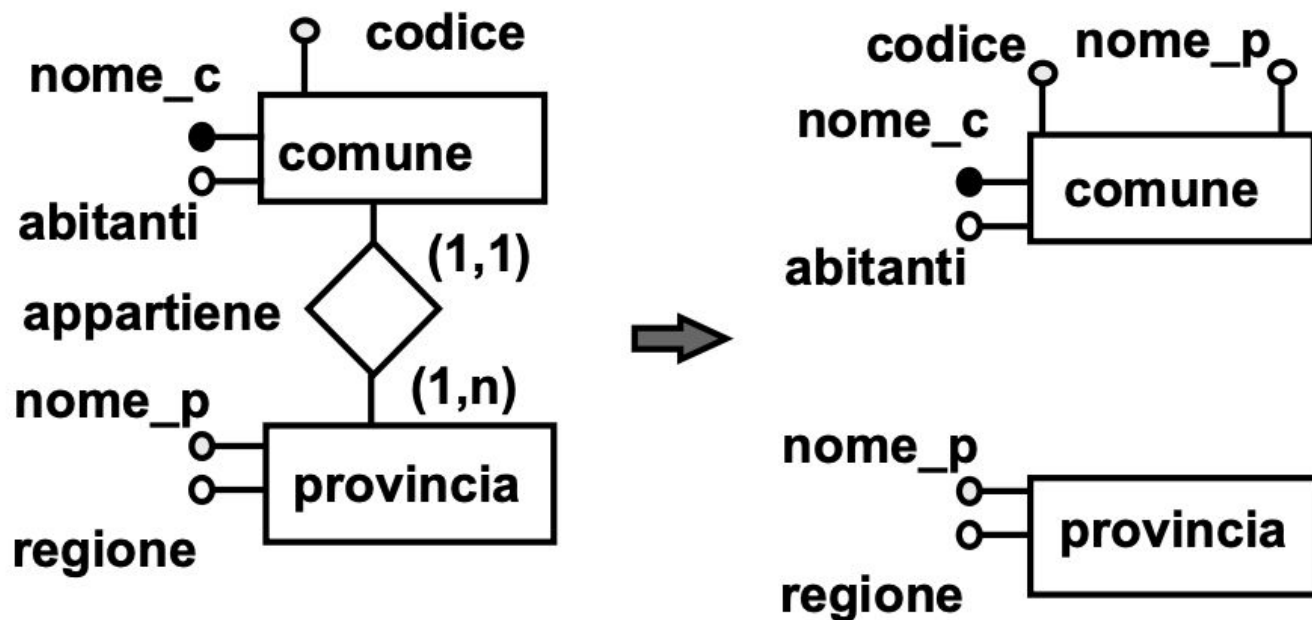
Traduzione standard



Altre traduzioni

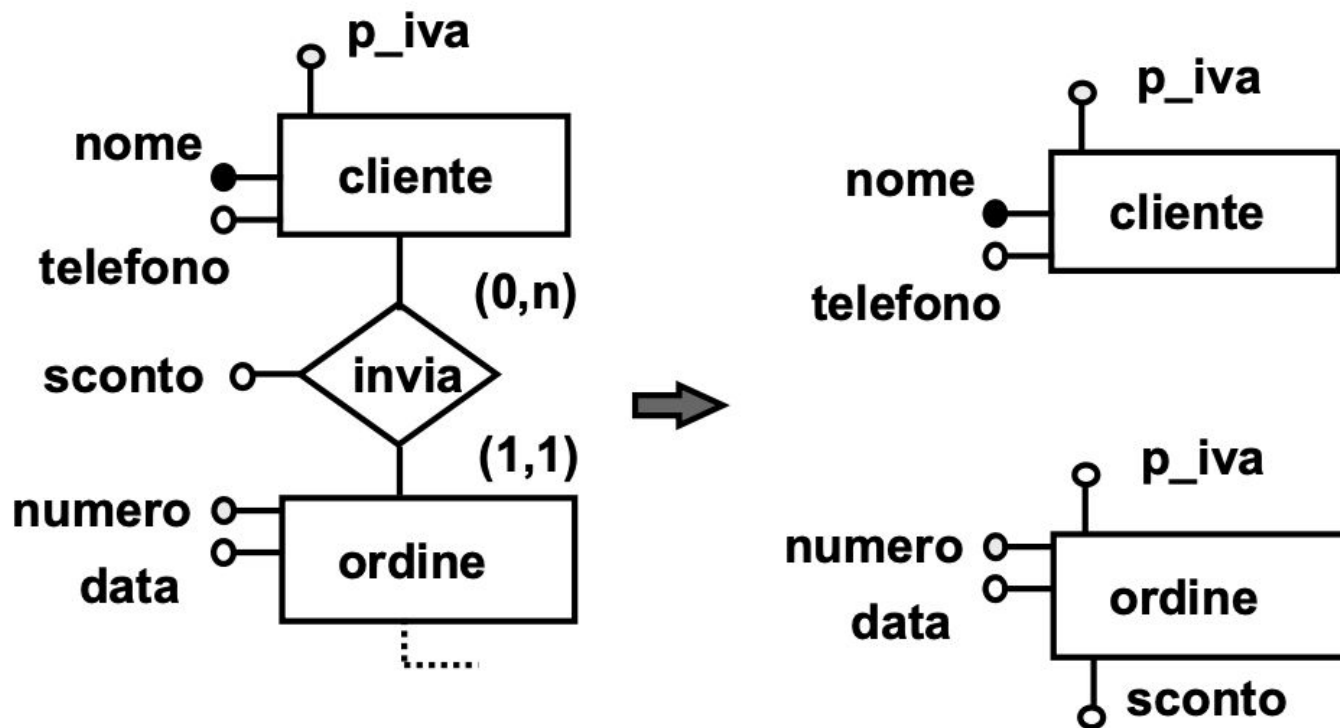
- La traduzione standard è sempre possibile ed è l'unica possibilità per le associazioni N a M.
- Altre forme di traduzione delle associazioni sono possibili per altri casi di cardinalità (1 a 1, 1 a N).
- Le altre forme di traduzione fondono in una stessa relazione entità e associazioni.

Ass. binaria 1 a N esempio



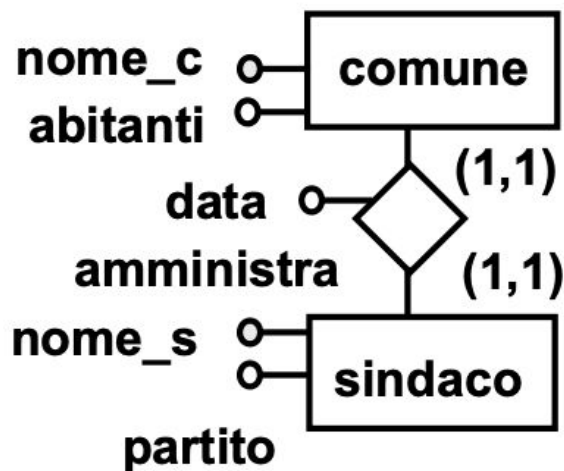
(senza attributi sull'associazione)

Ass. binaria 1 a N esempio



(con attributi sull'associazione)

Associazione binaria 1 a 1



- Traduzione con una relazione:

E12 (K1, A1, B1,
K2, A2, B2,
AR, BR)



Esercizi

Esercizio 1

Si deve progettare una base di dati relativa alla gestione delle prenotazioni dei posti di un laboratorio didattico di una università. Ogni studente è caratterizzato dalla propria matricola, nome, cognome, data e luogo di nascita, residenza, recapito telefonico. Gli studenti frequentano alcuni laboratori didattici. I laboratori didattici contengono un insieme di posti di lavoro e un insieme di risorse. Ad ogni posto di lavoro sono assegnate alcune risorse (unità di calcolo, stampanti, applicazioni). Alcune delle risorse sono rese disponibili a tutti gli studenti senza controlli, altre vengono assegnate agli studenti che frequentano determinati laboratori, previa autorizzazione. Lo studente può utilizzare un posto di lavoro solo se effettua una prenotazione. Si deve tenere traccia di tutte le prenotazioni e di tutte le volte che lo studente utilizza un posto di lavoro. Ogni laboratorio ha un solo responsabile, il quale si può occupare di un solo laboratorio.

Esercizio 2

Definire uno schema ER che descriva i dati di una applicazione relativa ad una compagnia aerea, secondo le specifiche che seguono.

- Ogni aeroporto ha un codice (che lo identifica univocamente), un nome, una città e una nazione (entrambe semplicemente indicate dal nome).
 - Una tratta è relativa a due aeroporti (indica in pratica il fatto che possono esistere voli dal primo al secondo aeroporto) e ha una lunghezza e un tempo di volo. Due tratte che siano una l'inverso dell'altra (ad esempio Milano-New York e New York-Milano) hanno la stessa lunghezza, ma possono avere tempi di volo diversi.
 - Un volo (inteso come concetto generico, ripetibile in giorni diversi) ha un numero identificante, collega due aeroporti, è associato ad un tipo di aeroplano e ha un orario di partenza e uno di arrivo (calcolabile sulla base dell'orario di partenza e del tempo di volo).
 - Per ogni tipo di aeroplano interessano il codice (identificante), il nome e il numero di posti.
- Modificare lo schema prodotto in risposta alle domande precedenti, tenendo conto delle seguenti specifiche aggiuntive:
- Per ogni tratta esistono uno o più tipi di aeroplano abilitati a servirla e il tempo di volo non è fisso ma dipende dal tipo di aeroplano.
 - Per ogni volo e per ciascuno dei giorni in cui è stato effettuato, interessa mantenere traccia dell'effettivo orario di partenza e di arrivo (che possiamo assumere differiscano sempre da quelli previsti) e dell'aeroplano con cui è stato effettuato, che deve appartenere al tipo previsto. Ogni aeroplano ha un tipo, un numero identificativo e un nome.

Esercizio 3

Si deve costruire una base di dati che descriva le informazioni relative al campionato di calcio. Per ogni partita descrivere il girone e la giornata in cui si é svolta, il numero progressivo nella giornata (es. prima partita, seconda partita, ecc.), la data con giorno, mese, anno, le squadre coinvolte nella partita con nome, città della squadra e allenatore, e infine per ciascuna squadra se ha giocato in casa. Si vogliono conoscere i giocatori che giocano in ogni squadra con la loro data di nascita e il loro ruolo principale. Si vuole conoscere per ogni giornata quanti punti ha ogni squadra. Si vogliono anche conoscere, per ogni squadra, i giocatori di ogni squadra che hanno giocato e in che ruolo ha giocato ogni giocatore (i ruoli , perciò, possono cambiare di partita in partita). Per ogni partita si vuole rappresentare l'arbitro con nome, cognome, città e regione di nascita. Distinguere le partite giocate regolarmente da quelle rinviate. Per quelle rinviate rappresentare la data in cui sono effettivamente giocate. Distinguere anche le partite giocate in una città diversa da quella della squadra ospitante; per queste si vuole rappresentare la città in cui si svolgono, nonché il motivo della variazione di sede. Dei giocatori interessa anche la città e la regione di nascita.

Esercizio 4

Realizzare uno schema ER relativo alla seguente descrizione: Una ditta produttrice di applicazioni software deve realizzare una base di dati relativa al personale, alle applicazioni prodotte ed ai clienti. Di ogni persona sono noti i dati anagrafici e la residenza. Il personale è diviso in programmatori, amministratori, venditori e dirigenti. Alcuni programmatori lavorano allo sviluppo delle applicazioni mentre altri forniscono un supporto ai clienti; vi sono anche programmatori che rivestono entrambe le funzioni. Le applicazioni si dividono in applicazioni finite per cui esiste un prezzo di listino, applicazioni in stadio beta (che vengono distribuite ad alcuni dei clienti in modo tale che vengano scoperti gli errori), applicazioni in stadio alfa (che sono nella prima fase dello sviluppo e che non vengono distribuite all'esterno). I clienti possono essere persone fisiche (caratterizzate dalle informazioni anagrafiche e dal codice fiscale) o ditte (caratterizzate dalla ragione sociale, dalla sede e dalla partita IVA). I clienti possono avere un contratto di manutenzione, per cui gli viene assegnato un particolare programmatore che fornisce assistenza quando sorgono problemi, oppure possono essere privi del contratto ed in tale caso ogni richiesta di intervento viene fatturata a parte e gestita da uno qualsiasi dei programmatori disponibili.



Domande?

masciadri.andrea@gmail.com

Ora provate voi..