

Appunti Basi di Dati

Luca Seggiani

17 Aprile 2024

1 Concetti inesprimibili nel modello ER

Alcuni concetti sono inesprimibili attraverso il modello ER tradizionale, e bisognano quindi di costrutti particolari:

- **Cardinalità**

- di relationship;
- di attributo;

- **Identificatore**

- interno;
- esterno;

- **Generalizzazione**

Cardinalità di relationship

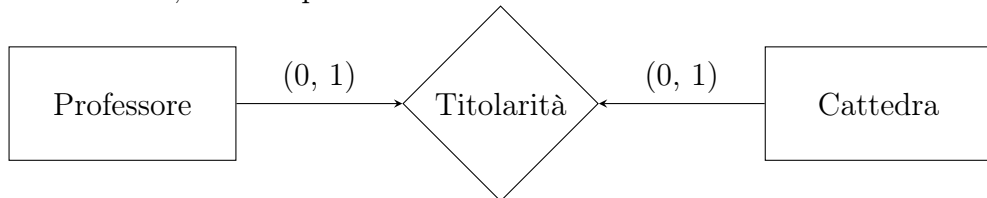
Una cardinalità di relationship è rappresentata da una coppia di valori associati ad ogni entità che partecipa alla relationship. Questi specificano il numero minimo e massimo di occorrenze della relationship a cui ciascuna occorrenza di entità può partecipare. I simboli usati saranno:

- Per la minima:
 - 0 \rightarrow partecipazione opzionale
 - 1 \rightarrow partecipazione obbligatoria
- Per la massima:
 - 1 \rightarrow partecipazione opzionale

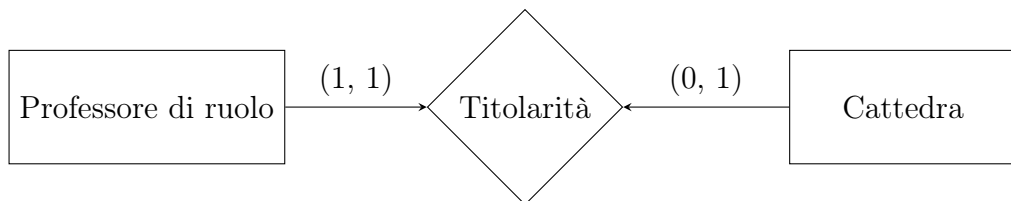
- $N \rightarrow$ non pone alcun limite

Con riferimento alle cardinalità massime, abbiamo relationship:

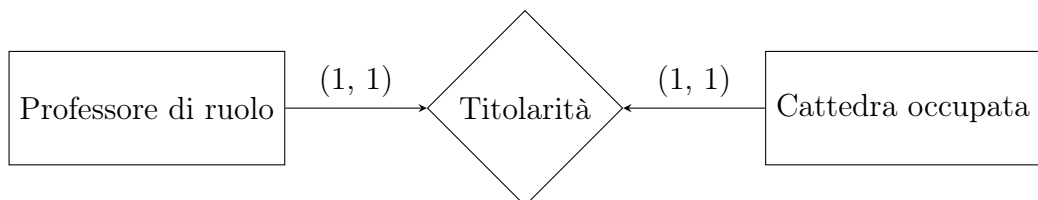
- **Uno a uno**, ad esempio:



Un professore può essere tale senza essere titolare di alcuna cattedra, e una cattedra può restare vuota.

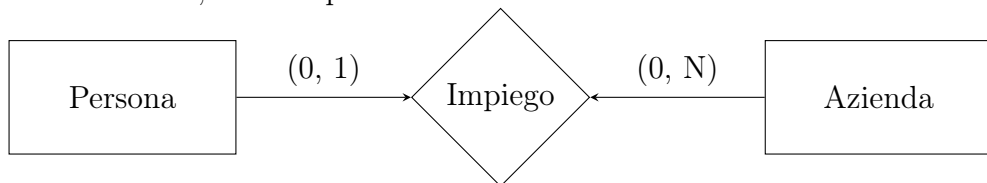


Un professore di ruolo dovrà ovviamente essere titolare di almeno una cattedra, ma questo non significa comunque che tutte le cattedre siano occupate.

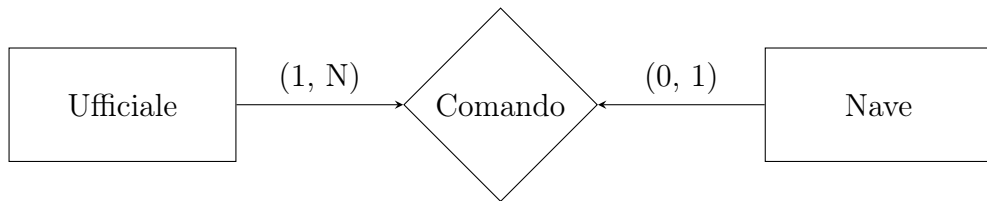


E' una forzatura? Sì. Serve a spiegare il concetto? Sempre sì.

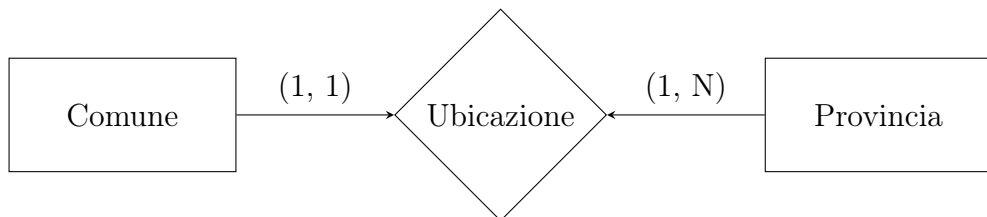
- **Uno a molti**, ad esempio:



Una persona può essere o non essere assunta, ma al massimo da una azienda. Al contrario, un'azienda avrà probabilmente più di un'impiegato.

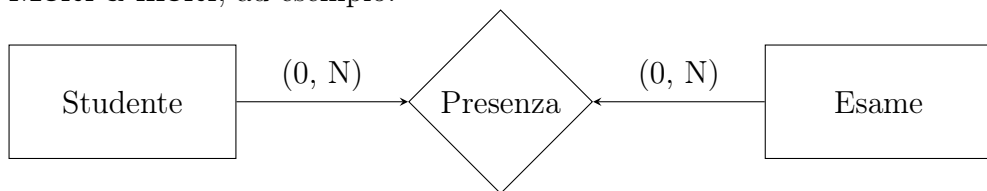


Un ufficiale per considerarsi tale deve comandare almeno una nave, ma non è detto che tutte le navi abbiano un comandante. Alcune sono navi fantasma come nei Pirati dei caraibi.

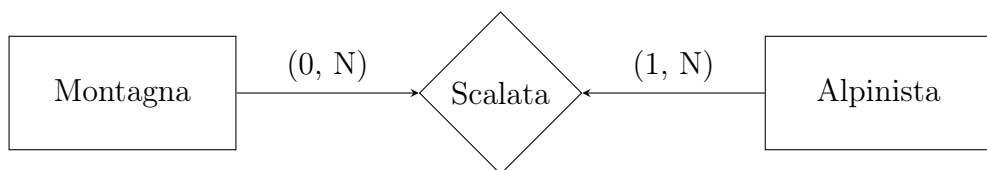


Ogni comune è ubicato in una e una sola provincia, mentre ogni provincia è l'ubicazione di più comuni.

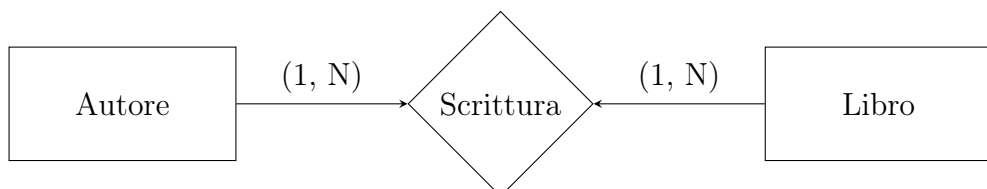
- **Molti a molti**, ad esempio:



Non è detto che ogni studente abbia sostenuto un'esame, come non è detto che ogni esame sia stato sostenuto da almeno uno studente.



Per potersi considerare alpinisti occorre aver scalato almeno una montagna, ma non è detto che tutte le montagne siano state scalate.

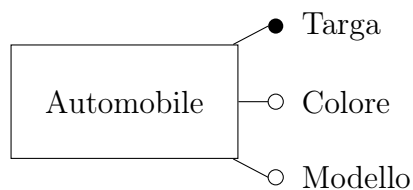


Ogni autore ha scritto almeno un libro per considerarsi tale, e ogni libro deve essere stato scritto da almeno un autore. Questo però non pregiudica che un autore non possa scrivere più libri o un libro non possa essere scritto da più autori.

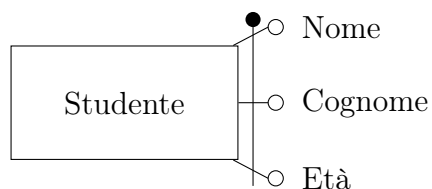
Identificatore di entità

L'identificatore di entità è uno strumento per l'identificazione univoca delle occorrenze di un'entità.

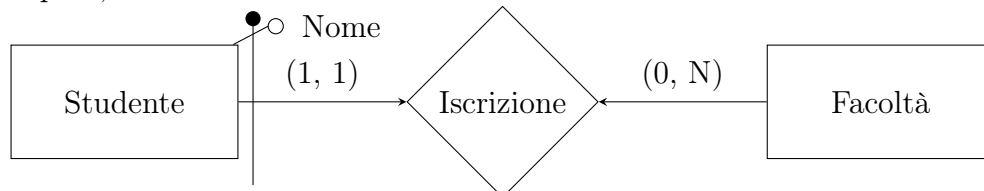
- Gli attributi dell'entità possono formare l'**identificatore interno** (o chiave). Per capirsi, un'identificatore interno è una chiave primaria o comunque una chiave candidata.



Come vediamo dalla figura, targa è la chiave primaria, ovvero l'identificatore. Nel caso un'insieme di attributi sia una chiave, adottiamo la rappresentazione seguente:



- Gli attributi dell'ent  e l'identificatore interno di ent  esterne raggiunte attraverso relationship formano un'**identificatore esterno**. un'identificazione esterna pu  essere possibile solo nel caso esista una relationship in cui l'ent  da identificare abbia cardinalit  (1,1). Per capirsi, un identificatore   una chiave contenente una chiave esterna.

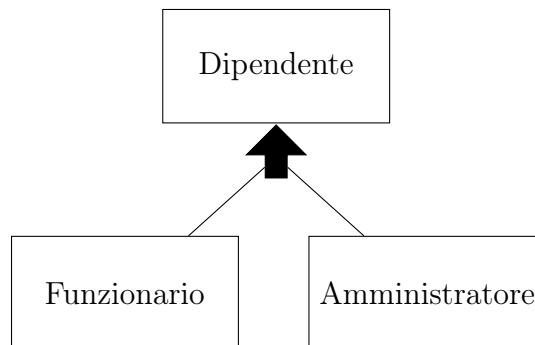


Dalla figura si capisce che il nome dello studente e la sua iscrizione a una certa facolt  formano il suo identificatore esterno.

Ogni entità deve possedere almeno un identificatore, ma può averne in generale più di uno.

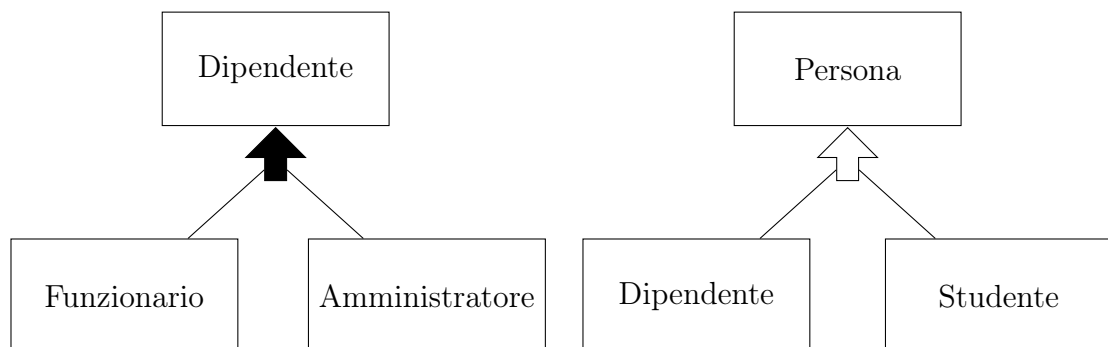
Generalizzazione

La generalizzazione mette in relazione una o più entità E_1, E_2, \dots, E_n con una singola entità E che le comprende come casi particolari. Si dice che E è una generalizzazione (oppure entità genitrice, madre) di E_1, E_2, \dots, E_n (dette specializzazioni, sottotipi o entità figlie).



Le caratteristiche delle generalizzazioni sono:

- **Ereditarietà:** tutte le proprietà dell'entità genitore vengono ereditate dalle figlie e non rappresentate esplicitamente.
- **Generalizzazione totale:** se ogni occorrenza dell'entità genitore è rimpiazzata da almeno un'occorrenza delle entità figlie, altrimenti si parla di **generalizzazione parziale**. Di norma le generalizzazioni parziali si indicano con una freccia vuota, mentre quelle totali con una freccia piena.



- **Generalizzazione esclusiva:** se ogni occorrenza dell'entità genitrice è occorrenza di al massimo una delle entità figlie, altrimenti si parla di **generalizzazione sovrapposta**.

- Possono esistere **gerarchie a più livelli** e multiple generalizzazioni allo stesso livello.
- Un'entità può essere inclusa in più gerarchie, sia come genitore che come figlia, se non entrambe.
- Se una generalizzazione ha solo un'entità figlia si dice **sottoinsieme**.
- Il genitore di una generalizzazione totale può non avere identificatore (**anonimato**) purché siano identificate le figlie.

2 Progettazione concettuale

Vediamo adesso la progettazione concettuale nel dettaglio. Il modello ER sarà lo strumento fondamentale nel corso di questa fase: dovremo innanzitutto decidere, per una qualsiasi specifica fornitaci, quale sia il costrutto ER più adeguato da utilizzare. Per fare questo ci basiamo sulle definizioni dei costrutti del modello ER:

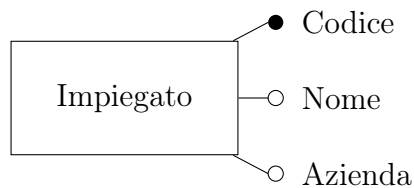
- **Entità**, se un oggetto ha proprietà significative e descrive oggetti con esistenza autonoma;
- **Attributo**, se un'oggetto è semplice e non ha proprietà specifiche;
- **Relationship**, se un concetto correla più oggetti;
- **Generalizzazione**, se un concetto è caso particolare di un altro.

Design pattern

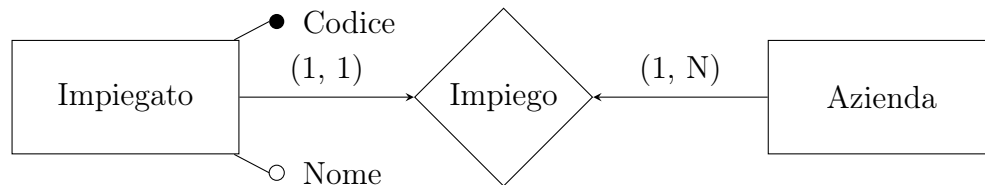
I design pattern sono soluzioni progettuali a problemi comuni. Sono largamente usati nell'ingegneria del software, e ne esistono alcuni comuni nella progettazione delle basi di dati.

- **Reificazione di attributo di entità**

La reificazione di un attributo di un'entità è la promozione di tale attributo ad un'entità a sé stante, con ovvia creazione di una relationship che esprima il rapporto fra l'entità di partenza e l'attributo che abbiamo trasformato in entità. Supponiamo di avere il modello ER:



L'attributo azienda potrebbe essere trasformato in un'entità a sé, per poter includere maggiori informazioni rispetto all'azienda stessa:



- **Reificazione di relationship in entità**

La reificazione di una relationship in entità è la promozione di una relationship in un'entità a sé stante, che sarà opportunamente collegata alle entità che metteva prima in relazione fra di loro attraverso altre relationship ausiliarie. Nello specifico, vediamo come usare la reificazione di relationship in entità per reificare relationship binarie e ternarie:

- **Reificazione di relationship binarie**

Diciamo di avere un modello abbastanza semplice, contenente due entità con identificatore interno legate da una relationship: Magari la relationship potrebbe avere bisogno di più informazioni specifiche: potremo allora usare più approcci alternativi. Il primo sarà di creare un'entità corrispondente alla relationship, e di rendere poi gli attributi di quell'entità e le relazioni con le entità di partenza identificatore esterno. Un'altro approccio sarà quello di assegnare semplicemente alla nuova entità un identificatore univoco.

- **Reificazione di relationship ternarie**

Un problema più sostanziale può essere rappresentato da una relationship ternaria. In questo caso si può adottare un approccio analogo al precedente, trasformando la relationship in un'entità contenente tutte le informazioni necessario e legata alle entità di partenza con 3 distinte relationship, che potranno anche formare l'identificatore esterno della nuova entità.

- **Reificazione di attributo di relationship**

Nel modello ER, nessuno ci impedisce di assegnare attributi alle relationship. Questo però andrà molto probabilmente tradotto in un'entità a sé stante, che sarà legata attraverso le metodologie che abbiamo appena visto alle entità che legava in partenza.

- **"Parte di"**

Il pattern parte di permette di definire concetti che sono parte di altri

concetti, come ad esempio un cinema con le sue sale. Viene rappresentato da una relazione "composizione", uno a molti, dove l'oggetto che è parte dell'altro ha identificatore esterno nel primo.

- **"Istanza di"**

Il pattern istanza di permette di definire concetti che sono istanze di altri concetti generali, come ad esempio un torneo e le sue edizioni. In questo caso si usa sempre una relazione, "occorrenza", uno a molti, dove l'oggetto occorrenza ha identificatore esterno nell'oggetto generale.

- **Caso particolare di entità**

Il pattern caso particolare di entità permette definire casi particolari di determinate entità, attraverso la generalizzazione totale di un'entità specializzata su un'entità generale. L'entità specializzata sarà il caso particolare.

Grafici di modelli ER per i pattern riportati si possono trovare al link: <https://github.com/Guray00/IngegneriaInformatica/blob/master/PRIMO%20ANNO/II%20SEMESTRE/Basi%20di%20dati/Diapositive/Slides%20Tonellotto/A.A.%2022-23/%5B06%5D%20Progettazione%20Concettuale%20e%20Logica.pdf>

Documentazione associata agli schemi concettuali

Il modello ER non basta mai da solo a descrivere l'interezza della realtà che vogliamo modellare. Esistono infatti, nonostante tutte le aggiunte che abbiamo apportato al modello ER, vincoli inesprimibili, che dovranno essere espressi dalla documentazione di supporto. Serviranno quindi:

- Un **dizionario dei dati**, ovvero una tabella che contenga informazioni riguardo alle entità e alle relationship che dovranno essere modellizzate dal nostro database.
- **Regole aziendali**: ovvero informazioni riguardo ai vincoli di integrità necessarie, nonché possibili derivazioni specifiche ai tipi di dati che staremo trattando (formule particolari per il calcolo di indici, medie...).