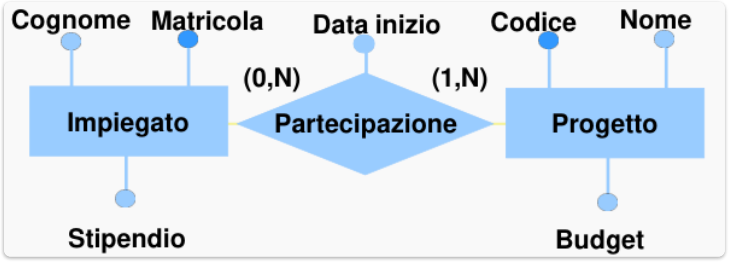
CAP09_TRADUZIONE

Table of contents

Come tradurre lo schema E-R in schema Logico Relazionale

Le *entità* diventano delle relazioni sugli stessi attributi, le *associazioni* (non tutte) diventano tabelle/relazioni sulle entità coinvolte e se hanno degli attributi, questi vengono inseriti.

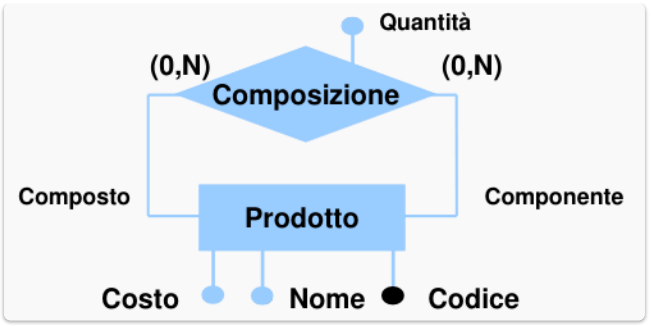
Molti a molti (N, N)



I nostri ID delle relazioni, sono le *chiavi primarie* delle tabelle: nel caso sopra, `Partecipazione` diventa una *tabella* con chiavi `Matricola` e `Codice` delle relazioni a cui si associa `Impiegato` e `Progetto`.

`Impiegato` (Matricola, Cognome, Stipendio)
`Progetto` (Codice, Nome, Budget)
`Partecipazione` (Matricola, Codice, DataInizio)

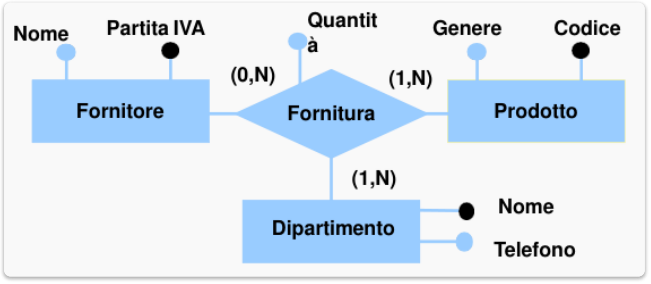
Siamo felici con la traduzione?
Non proprio: vincoli di *chiave esterna* sarebbero migliori piuttosto che relazione con 2 chiavi che gia' esistono altrove, magari cambiando anche i nomi per renderli più espressivi. Inoltre, nell'associazione dal lato cardinalità N, non riusciamo a tenere conto delle cardinalità minime della relationship.



In questo caso l'associazione e' ricorsiva: fissato un prodotto abbiamo N altri prodotti (`Composto`), oppure fissato un prodotto, lui è componente di altri (`Componente`).

`Prodotto` (Codice, Nome, Costo)
`Composizione` (Composto, Componente, Quantità)

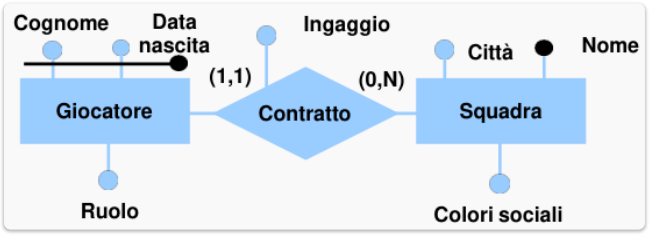
Per scrivere che la nostra implementazione è ricorsiva, cambiamo il nome della chiave primaria di `Composizione`.



La traduzione segue le stesse meccaniche, solo che stavolta abbiamo una tripla di chiavi esterne.

`Fornitore` (PartitaIVA, Nome)
`Prodotto` (Codice, Genere)
`Dipartimento` (Nome, Telefono)
`Fornitura` (Fornitore, Prodotto, Dipartimento, Quantità)

Uno a molti $(1, N)$



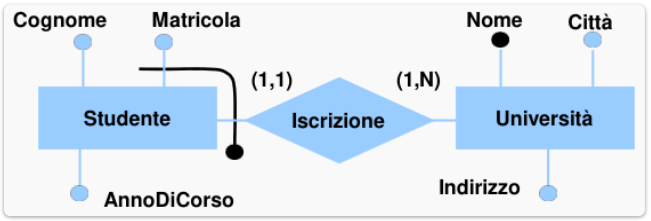
Per implementare la cardinalità uno a molti, un `Giocatore` non può avere più di un contratto. La relazione va forzata, uno e un solo `Contratto` dentro la relazione `Giocatore`, la cardinalità minima di 1 la imponiamo in questo modo:

`Giocatore` (Cognome, DataNasc, Ruolo, Squadra, Ingaggio)
`Squadra` (Nome, Città, ColoriSociali)

Quindi l'associazione viene eliminata e i nostri attributi inseriti nel lato molti `Giocatore`. Inoltre, se la cardinalità minima è:

- 0 allora il valore nullo(*) è *ammesso*;
- 1 allora il valore nullo(*) *non è ammesso*.

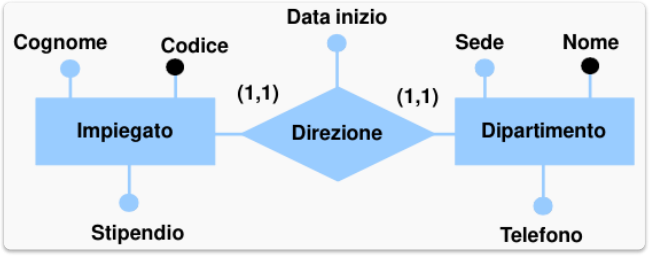
Con identificatore esterno



Quando l'identificatore esterno c'è, la cardinalità massima diventa allora *uno a uno* $(1,1)$, se cosi' non fosse, errori concettuali nello schema sono presenti.
Il nome dell'`Universita` entra come chiave in `Studente`.

`Studente` (Matricola, Università, Cognome, AnnoDiCorso)
`Universita` (Nome, Città, Indirizzo)

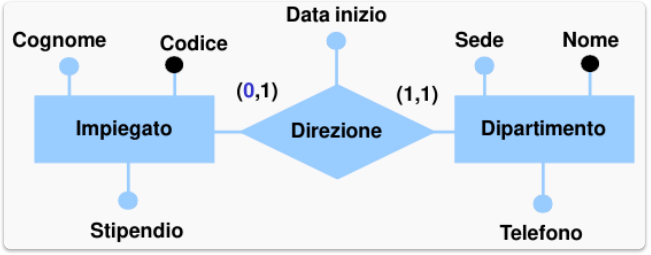
Uno a uno $(1,1)$



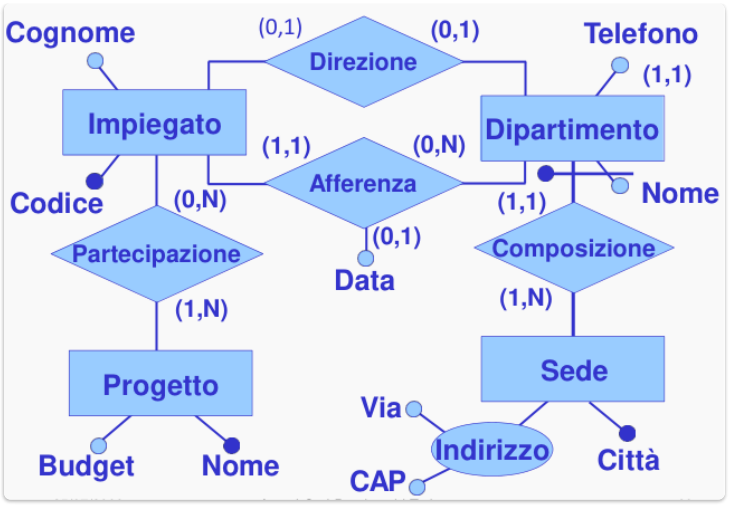
Come visto sopra, quando abbiamo una cardinalità massima 1, non creiamo una tabella, ma spostiamo gli attributi in una relazione... ma su quale lo facciamo in questo caso che abbiamo in ambo lati 1?
Usiamo la logica che se per caso su uno dei due lati avessimo 0, allora la traduzione sarebbe:

`Impiegato` (Codice, Cognome, Stipendio)
`Dipartimento` (Nome, Sede, Telefono, Direttore, InizioD)

E se non volessimo lo stesso `Direttore` per più `Dipartimenti`?
Codifichiamo con `UNIQUE` su `Direttore`, forzando l'associazione ad $1,1$.



Esempio di traduzione



Impiegato (Codice, Cognome, Dipartiemento, Sede, Data*)
Dipartimento (Nome, Città, Telefono, Direttore*)
Sede (Città, Via, CAP)
Progetto (Nome, Budget)
Partecipazione (Impiegato, Progetto)