



PEGASO
Università Telematica



Indice

1. INTRODUZIONE.....	3
2. I PATTERN.....	4
3. REIFICAZIONE ATTRIBUTO.....	5
4. RELAZIONI UNO-A-MOLTI	6
4.1 PATTERN PART-OF.....	6
4.2 PATTERN INSTANCE-OF.....	7
5. RELAZIONE MOLTI-A-MOLTI.....	8
6. RELAZIONE RICORSIVA.....	10
BIBLIOGRAFIA	12

1. Introduzione

Questa unità didattica di apprendimento illustra i pattern di progettazione o design patterns, ovvero una serie di soluzioni per problemi che si incontrano comunemente nella progettazione di una base dati. Nella prima sezione si chiarisce tale concetto.

Nella seconda sezione si parla applicazione di pattern ad attributi.

Nelle altre sezioni si illustrano le regole di applicabilità dei pattern a problematiche comuni, guidati dalla chiave di lettura rappresentata dal tipo di relazione coinvolta.

Così si illustrano con esempi i pattern relativi alle relazioni uno-a-molti, molti-a-molti e ricorsive.

2. I pattern

In questa sezione si introducono i pattern, ovvero un concetto che può essere definito “*una soluzione progettuale generale ad un problema ricorrente*”.

Si tratta di una descrizione o modello logico da applicare per la risoluzione di un problema che può presentarsi in diverse situazioni durante le fasi di progettazione, sia a livello di progettazione del software sia a livello di progettazione di una base dati.

Infatti, in molti schemi E-R si ritrovano dei “pattern” comuni, ovvero soluzioni a problemi di progettazione di basi di dati che si presentano di frequente.

Vediamo adesso alcuni pattern comuni nella progettazione concettuale di basi di dati, iniziando con una loro classificazione che ci guiderà nel nostro studio.

- **Classificazione generale dei pattern**
 - *Pattern relativi ad attributi*
 - Reificazione attributo
 - *Relazioni uno-a-molti*
 - Part-of
 - Instance-of
 - *Relazioni multi-a-molti*
 - *Relazioni ricorsive*

3. Reificazione attributo

Iniziamo con il caso più semplice, ovvero il caso riguardante un attributo. Quando si individua nelle specifiche un concetto autonomo con proprietà associate, come attributo di una entità e si vuole dare più peso alle informazioni rappresentate da tale attributo. Ad esempio, per l'entità impiegato, si ha l'attributo di Azienda, come rappresentato in Figura 1. Però si vuole dare più valore a tale attributo.

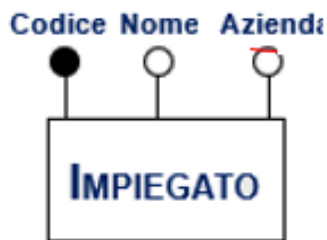


Figura 1: l'entità IMPIEGATO.

Per fare questo, si effettua la **Reificazione dell'attributo Azienda**. Nella rappresentazione precedente l'azienda è solo un attributo, ovvero una stringa che si assegna ad ogni occorrenza di impiegato. Se vogliamo però rappresentare esplicitamente il concetto di azienda, l'attributo azienda viene promosso ad entità, come illustrato in Figura 2.

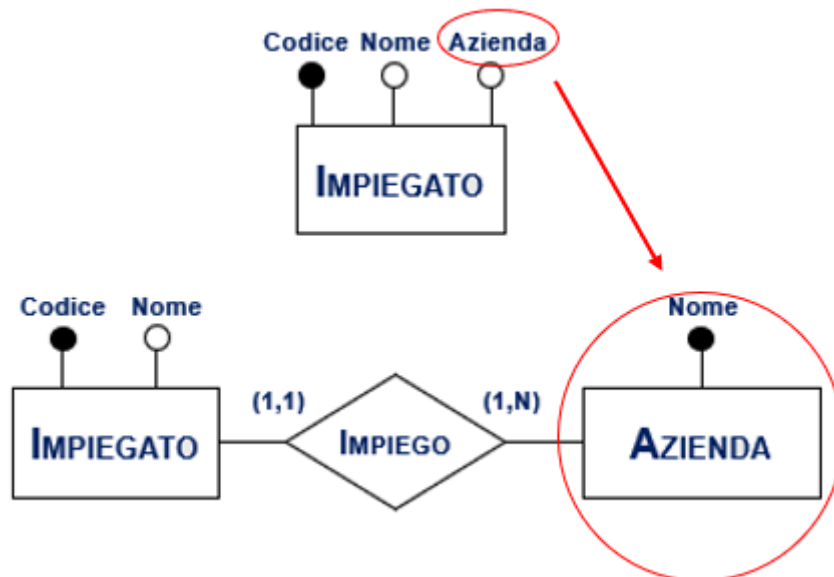


Figura 2: reificazione dell'attributo azienda.

4. Relazioni uno-a-molti

In questo capitolo studiamo i pattern che si applicano a relazioni uno-a-molti.

4.1 Pattern Part-of

Questo pattern si applica quando un'entità è parte di un'altra entità in una relazione uno-a-molti. Si possono presentare le seguenti situazioni:

1. Una sala di un cinema esiste solo perché parte del cinema, che può avere più sale, altrimenti **non esiste**.
2. Un tecnico ha una sua esistenza **indipendentemente** se appartiene o no ad un team.

Per la soluzione del Caso 1, si è dovuto mettere una identificazione esterna (Cinema) poiché l'entità Sala esiste solo se esiste l'entità Cinema, così come raffigurato in Figura 3.

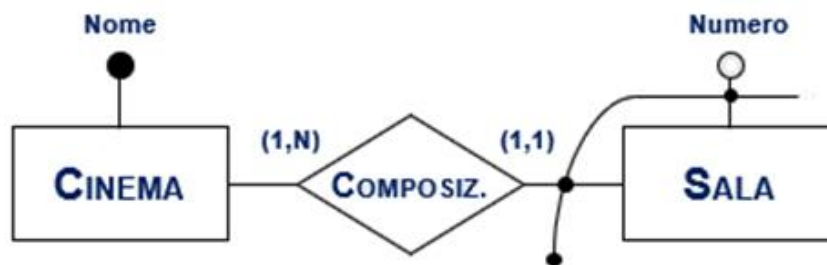


Figura 3: diagramma-soluzione del caso 1.

Per il caso 2, la Figura 4 ne presenta la soluzione. L'entità Tecnico esiste indipendentemente dall'appartenenza ad un team. In questo caso il tecnico ha un identificatore rappresentato da un codice.

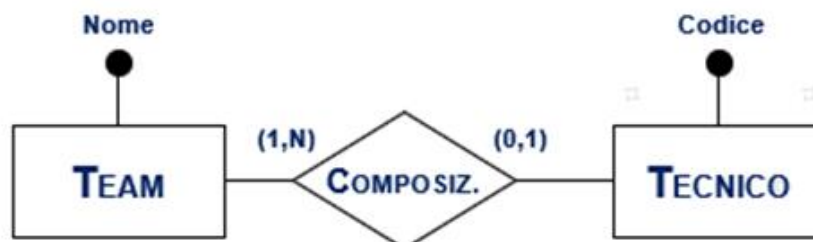


Figura 4: soluzione del caso 2.

4.2 Pattern Instance-of

Questo pattern si può applicare quando le istanze delle occorrenze di una entità di una relazione sono istanze delle occorrenze dell'altra entità. Come esempi abbiamo due scenari:

1. Entità astratta volo e entità volo reale. Nel caso di un volo tipo AZ234 che parte ogni giorno.
2. Entità torneo e entità edizione del torneo. Un torneo che ogni anno viene svolto.

Per quanto riguarda la soluzione al caso 1, facciamo riferimento alla Figura 5, dove l'identificazione del volo avviene tramite la data e, esternamente, il volo di cui è istanza. Infatti, per individuare un volo, è necessario conoscere anche la data oltre al codice del volo stesso.

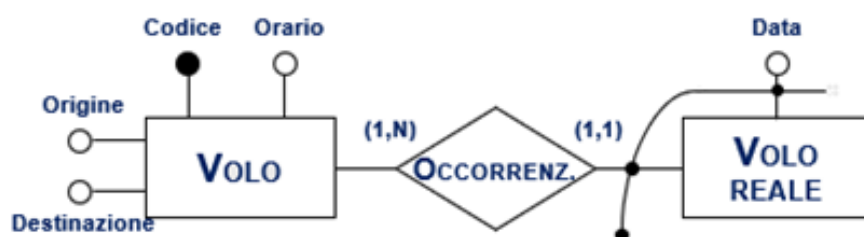


Figura 5: schema soluzione al caso 1.

La soluzione al caso 2 è quella illustrata in Figura 6. Anche in questo caso, l'entità «Edizione Torneo» ha un identificatore esterno, riferito all'entità «Torneo».

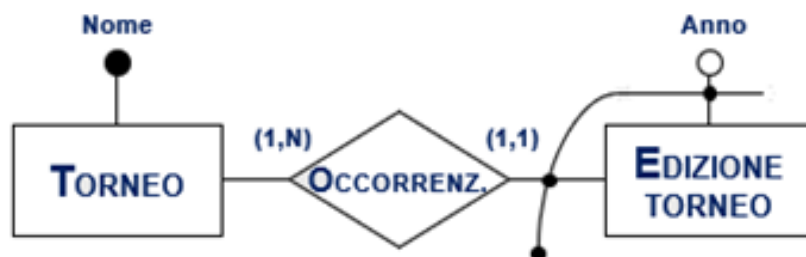


Figura 6: soluzione al caso 2.

5. Relazione multi-a-molti

In questa sezione studiamo i pattern relativi alle relazioni multi-a-molti. Si tratta di una reificazione di relazione binaria, quando si ha un concetto che lega altri due concetti. Come esempio vediamo il caso di uno studente che dà l'esame di un corso.

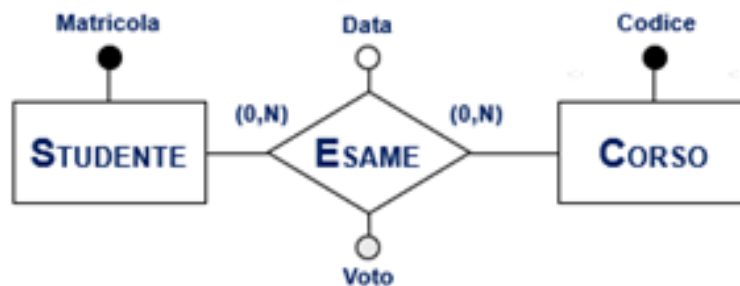


Figura 7: esempio di relazione multi-a-molti.

Uno studente può dare però due volte lo stesso esame e non è possibile rappresentare tale situazione con lo schema proposto (problema delle coppie duplicate) dalla Figura 7. La soluzione consiste nell'applicare la Reificazione della relazione Esame, come raffigurato in Figura 8. In pratica, la relazione Esame è stata promossa ad entità e sono state inserite due relazioni: la relazione Studente-Esime (S-E) e la relazione Esime-Corso (E-C). L'entità Esime ha quindi un identificatore interno, la Data e due identificatori esterni, lo Studente e il Corso.

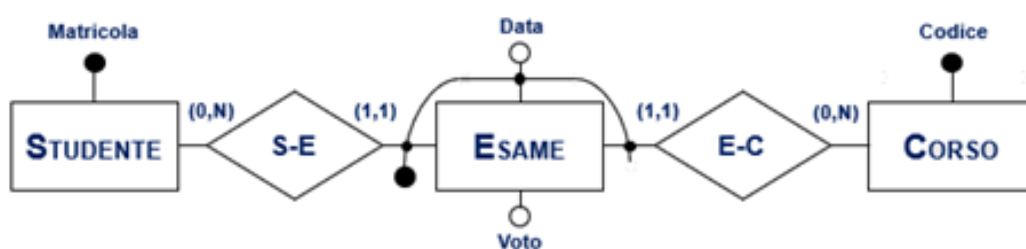


Figura 8: reificazione della relazione Esame.

Una soluzione alternativa a quella appena esposta è quella illustrata in Figura 9. Abbiamo introdotto un **codice** come identificatore che ha reso la soluzione più semplice. Si ponga attenzione al fatto che comunque il codice non era però nelle specifiche.

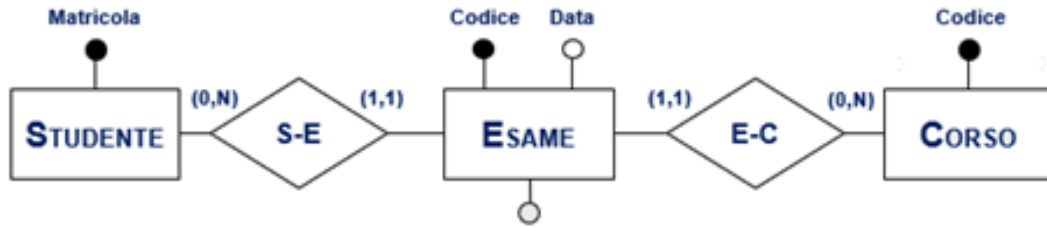


Figura 9: alternativa al caso di studio.

6. Relazione ricorsiva

In questo capitolo vediamo un pattern che riguarda le relazioni ricorsive. Ricordiamo che una relazione la diciamo ricorsiva se è una relazione tra una entità e sé stessa, come illustrato nelle Figure 10 e 11.

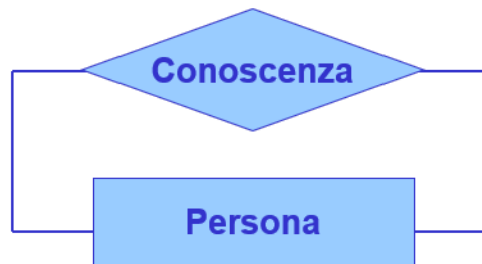


Figura 10: esempio di relazione ricorsiva semplice.

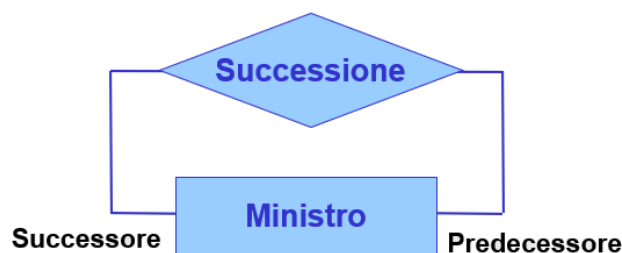


Figura 11: esempio di relazione ricorsiva con successore e predecessore.

Per comprendere questo pattern, ovvero la reificazione di relazione ricorsiva, rappresentiamo una partita di calcio tra due squadre. Il concetto di partita può essere visto inizialmente come una relazione ricorsiva sull'entità Squadra dove le stesse squadre si incontrano però più volte.

La soluzione è quella illustrata in Figura 12, dove si assume che una squadra non possa giocare due partite nello stesso giorno. La relazione Partita è stata reificata ad entità e sono state aggiunte due relazioni, Casa e Ospite con le loro cardinalità. L'identificazione dell'entità Partita coinvolge solo la data e la squadra che gioca in casa.

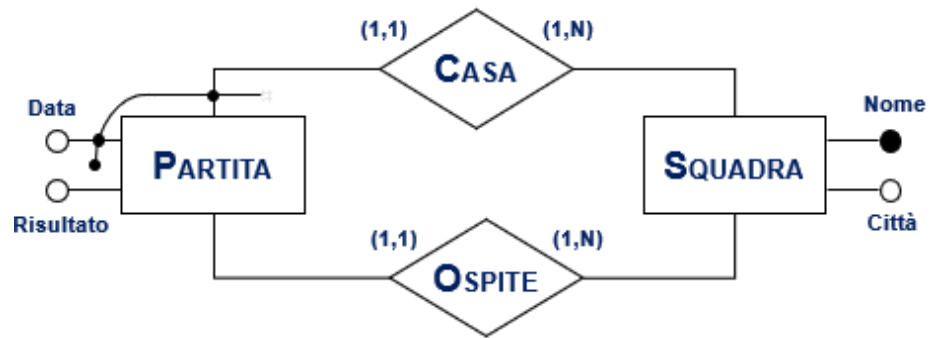


Figura 12: reificazione di una relazione ricorsiva.

Bibliografia

- Atzeni P., Ceri S., Fraternali P., Paraboschi S., Torlone R. (2018). Basi di Dati. McGraw-Hill Education.
- Batini C., Lenzerini M. (1988). Basi di Dati. In Cioffi G. and Falzone V. (Eds). Calderini. Seconda Edizione.