

Qualche nozione sui database(prima di poter parlare di E/R)

Quando è necessario organizzare realtà complesse e realizzare un **sistema informativo automatizzato**, cioè la memorizzazione e gestione informatica di grandi quantità di informazioni (sistema EDP) , dobbiamo servirci dei Data Base.

Il passaggio dagli archivi di dati (files) alle basi di dati è motivato dai vantaggi che un DB presenta rispetto alle organizzazioni fisiche di piu' basso livello.

Per prima cosa individuiamo una definizione di DB :

una **Base di Dati** è un insieme di files **collegati** tra loro, memorizzati permanentemente in memoria di massa, per rappresentare una realtà, gestiti con un sw appropriato chiamato **DBMS** (DataBase Management System).

Quindi è il **collegamento** tra i diversi archivi la fondamentale differenza: se negli archivi tradizionali ogni file è indipendente dagli altri, in una base di dati ogni archivio viene visto in relazione agli altri. Inoltre tutta la gestione fisica della struttura (inserimenti, ricerche, cancellazioni ecc.) risulta a carico del DBMS e trasparente al programmatore (bella svolta :-).

La visione unitaria di piu' archivi, rispetto ad una frammentazione delle informazioni, consente di eliminare la duplicazione dei dati (**ridondanza**). Tale obiettivo è di importanza primaria: la duplicazione dei dati può infatti portare alla **inconsistenza** dei dati (dati non affidabili) per problemi di **incongruenza** (valori diversi per la stessa informazione): infatti un dato duplicato deve essere inserito in tutti gli archivi che lo contengono, ed in fase di modifica, aggiornato in tutti gli archivi.

Prima di poter realizzare sul computer il nostro db è necessaria una fase di progettazione preparatoria: si deve studiare la realtà da informatizzare per estrarne i dati da memorizzare, le loro relazioni e le loro caratteristiche. Questo passaggio viene definito **Modellazione dei dati**, e consiste nella costruzione di un modello astratto (**schema concettuale**) comprensibile anche ai non tecnici.

Lo schema concettuale piu' diffuso è il modello Entity/Relationship (modello **E/R**)

[**N.B.**: questo **NON** è un argomento solo teorico. Per poter effettivamente implementare un db sul tuo computer devi prima realizzare questo schema che poi tradurrai con l'uso del sw adeguato, quindi è **INDISPENSABILE** studiarlo!!!]

Modello E/R

Lo schema E/R rappresenta una modalità semplice ,ma estremamente efficace di modellazione dei dati. Il diagramma E/R ha un simbolismo grafico lineare ed immediatamente comprensibile, pur mantenendo il necessario standard di rigosità ed efficienza necessarie ad un progetto informatico.

Gli elementi di un modello E/R sono:

- Entità
- Attributi
- Associazioni

e a ciascuno di essi risponde un ben preciso simbolo grafico.

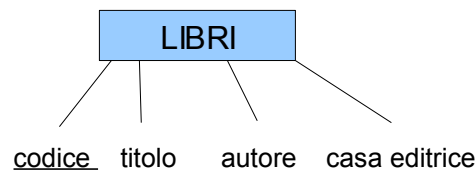
Entità

Le entità sono gli oggetti che dobbiamo rappresentare, cioè le nostre unità informative.
Se ad es. volessimo rappresentare una biblioteca avremmo l'entità libri, una scuola l'entità alunni ecc. Il simbolo che rappresenta un'entità è il rettangolo



Attributi

Gli attributi sono le caratteristiche di ogni entità, cioè le proprietà che le contraddistinguono.
Nell'entità libri potremmo ad es. individuare un codice, il titolo, l'autore, la casa editrice ecc.



L'attributo sottolineato viene definito **chiave primaria** (PK) ed è quel campo che individua in modo univoco ogni singolo elemento (istanza) di un'entità. Non può mai quindi contenere valori ripetuti.

Nel nostro esempio ogni libro ha un codice ben definito e non possono esistere due libri con il medesimo codice.

Caratteristiche di ogni attributo sono:

nome

formato (carattere, numerico, data/ora ecc.)

dimensione

opzionalità

vincoli di integrità

Per opzionalità si intende la possibilità o meno che un attributo non assuma valore (pensa ai campi obbligatori di una registrazione) mentre per vincoli di integrità, le eventuali restrizioni sul dominio, cioè sull'insieme dei valori ammessi (ad es. l'attributo votodiploma dell'entità studente ammette solo i valori tra 60 e 100). L'impostazione di questi vincoli, estremamente difficoltosa con gli archivi, in termini di istruzioni da scrivere, risulta tutta a carico del DBMS.

[Per rendere un po' più chiaro il tutto prova a sostituire, in modo poco teorico ma molto efficace la parola entità con il termine file, attributo con campo ed istanza con record!!!]

Associazioni

L'associazione è una relazione che unisce le entità, cioè la corrispondenza tra gli elementi delle entità stesse. Rappresenta quindi quel collegamento tra gli archivi che è la caratteristica principale del concetto di base di dati.

Il simbolo grafico per rappresentare una relazione è una linea che unisce le entità



In questo esempio si parla di **associazione 1 a 1**, ad ogni elemento della prima entità corrisponde uno ed un solo elemento della seconda entità e viceversa: ad ogni studente corrisponde solo il suo diploma.

Come puoi notare per stabilire una relazione è necessario che le due entità abbiano un campo comune che le due entità abbiano un campo comune, cioè con lo stesso significato. **Controlla sempre attentamente questo passaggio, in caso di errore infatti l'intero progetto viene invalidato!**



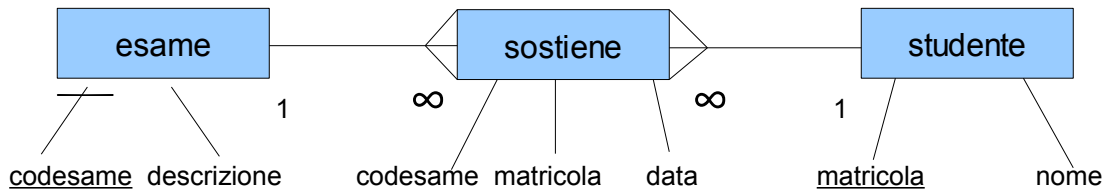
In questo esempio si parla di **associazione 1 a molti**, ad ogni elemento della prima entità corrispondono più elementi della seconda entità, mentre ad ogni elemento della seconda entità corrisponde un solo elemento della prima: ad ogni facoltà possono iscriversi più studenti, mentre un singolo studente può iscriversi ad una sola facoltà.

Anche in questo caso le due entità hanno un campo comune, cioè il "codicefacoltà", ma nella seconda entità non è una chiave primaria. In questo caso infatti i valori in esso contenuti possono ripetersi molte volte. In tal caso parliamo di **chiave secondaria**. **Ricorda che l'entità che ha come campo di collegamento la chiave secondaria ha sempre la cardinalità molti.**



In questo esempio invece la cardinalità della relazione è molti a molti: ogni esame viene sostenuto da diversi studenti ed ogni studente può sostenere diversi esami.

Questo tipo di rappresentazione non è implementabile, per questo viene trasformata in un modo che il nostro sw potrà poi rappresentare:



Viene creata una entità intermedia di collegamento tra le due che ci consente di sostituire alla complessità della relazione molti a molti una duplice corrispondenza: la prima di cardinalità 1 a molti, l'altra di cardinalità molti a 1.

Come puoi vedere questo schema si ottiene prendendo le chiavi primarie delle entità di partenza ed assegnandole come attributi, nella entità intermedia,.

Una entità viene definita **entità forte** quando le informazioni che rappresenta esistono anche senza le altre entità dello schema, viene invece definita **entità debole** quando ha necessità di essere relazionata ad un'altra entità per assumere un significato. Un'entità forte, al contrario di quella debole, non ha bisogno di altre entità per essere identificata. Nell'esempio precedente studente ed esame sono entità forti, mentre sostiene è un'entità debole, in quanto i suoi elementi esistono solo se gli studenti sostengono effettivamente gli esami.

Quando devi progettare una base di dati che necessiti di relazioni tra molte entità individua per prima cosa le entità forti, attorno ad esse poi potrai stabilire tutte le associazioni.

In alcune rappresentazioni potrai trovare la entità intermedia con il simbolo

