

Le basi del Data Base

Data Base Management Systems: i sistemi di gestione per basi di dati

di Andrea de Prisco

Prima parte

Immaginate di trovarvi in una moderna biblioteca, presso l'addetto ai prestiti e alle consultazioni.

Davanti a lui, a fosfori verdi, un terminale di un computer è pronto per qualsiasi interrogazione.

L'addetto dice: "Nome e Cognome, prego!"

Qualche secondo dopo: "Lei non ha mai usufruito di questa biblioteca, desidera consultare o richiedere in prestito un libro?"... "Vorrei in prestito per qualche giorno il Knuth, the art of computer programming, vol. I".

Pochi secondi ancora: "mi dispiace, è già in prestito, se vuole inserisco la sua prenotazione per questo testo; torni tra due o tre giorni, il possessore è già stato avvisato che è scaduto il tempo concessogli".

Tutto automatizzato: le prenotazioni, gli elenchi dei prestiti, degli utenti, dei materiali (libri e pubblicazioni), perfino ogni mattina, la stampa di tutti i testi che dovrebbero essere riconsegnati in giornata.

Dietro tutto questo, non solo un mega-computer dotato di sofisticate periferiche per l'archiviazione, ma una vera e propria base di dati.

A partire da questo numero, ci occuperemo un po' di Data Base, una parola che tutti avranno sentito almeno una volta, anche se non riguarda troppo da vicino il mondo della microinformatica.

Si sa che per gestire centinaia di migliaia di dati un "floppy" da personal è sicuramente ridicolo: guarderemo questo aspetto, forse da un punto di vista più professionale che hobbyistico, senza però entrare nel "difficile".

I sistemi di gestione per basi di dati hanno interessato e interessano parecchi studiosi del mondo.

È il vero e proprio mondo dell'informatica, che come scienza studia l'informazione automatica: ciò che maneggiava l'addetto di cui sopra.

Cosa è un Data Base

È opinione diffusa che l'informazione posseduta da una qualsiasi azienda può essere considerata, senza troppi sforzi, come una risorsa patrimoniale allo stesso modo di come tempo addietro poteva essere considerata risorsa il capitale, il lavoro, l'energia. Inoltre, il (buon) funzionamento di qualsiasi organizzazione è condizionato dall'accuratezza dell'informazione di cui dispone e dalla tempestività con cui viene elaborata per prendere decisioni.

I sistemi di gestione per basi di dati permettono, con una certa facilità, di trattare moli enormi di dati, per così dire, senza farsi prendere dal panico. Fino a qualche anno fa, le informazioni venivano archiviate in file di dati, ed era compito del programmatore curare l'organizzazione di questi su memoria di massa. In altre parole: grosso lavoro per produrre procedure automatizzate non molto dissimili, per complessità, a quelle manuali, e spesso meno affidabili. I Data Base possono essere visti come dei linguaggi di programmazione ad alto livello, particolarmente orientati a facilitare al massimo i trattamenti dei dati.

Fra sistemi di archiviazione e Sistemi di Gestione per Basi di Dati (SGBD), possiamo dire che intercorre la stessa differenza che c'è tra un linguaggio di programmazione tipo Assembler (non dotato di alcuna struttura pre-programmata) e un linguaggio ad alto livello, come il Pascal o il Basic.

In altre parole, coi SGBD il programmatore della base di dati, detto anche Data Base Administrator o DBA, è sgravato da buona parte dei compiti (quel che riguarda l'organizzazione fisica e logica dei dati) e può dedicarsi a trattare meglio le procedure che saranno utilizzate dall'utente. Notare come si stia parlando in termini tutt'altro che hobbyistici.

Prima di entrare in questo particolare mondo, bisognerebbe rinfrescare un po' le idee sul concetto di dato soprattutto considerando che la maggior parte dei lettori conoscono solo il linguaggio Basic...

Il dato

Chiunque abbia scritto un programma Basic, conoscerà la differenza che c'è tra un intero e un reale o tra un intero e una stringa. Per chi non lo sapesse, la differenza è appunto nel "tipo". Ciò significa che due oggetti di tipo diverso sono necessariamente diversi, anche se dovesse capitare che rappresentino la stessa roba. Per intenderci: il numero 123 è ben diverso dalla stringa "123", e non solo come organizzazione fisica all'interno del nostro calcolatore. Col primo possiamo fare somme, prodotti, sottrazioni, col secondo no. Questo perché l'operazione di somma è definita per funzionare tra due numeri e non tra due cose che hanno sembianze di numero.

Il Basic si ferma qui. Con linguaggi di programmazione un po' più evoluti, il tipo di dato è qualcosa di molto più flessibile, in alcuni casi se ne possono inventare di nuovi a partire dai tipi già definiti. Il primo tipo di dato che analizzeremo è l'ennupla. È noto che l'oggetto (3,4) rappresenta una coppia. Estendendo il nostro discorso a oggetti diversi da numeri, sarà una coppia anche ("Mario", "Rossi"). Dando un nome ad ognuno dei due campi, la coppia diventa, ad esempio, (Nome = "Mario", Cognome = "Rossi"). Questa è una ennupla formata da due campi di tipo stringa, Nome e Cognome. A partire dal tipo stringa, abbiamo costruito un nuovo "tipo".

Per fare un discorso un tantino più operativo, ci siamo inventati per voi un metalinguaggio di programmazione, chiaramente non implementato, che ha solo l'umile pretesa di mostrare come trattare nuovi tipi. Per qualche affinità col più diffuso dei linguaggi di programmazione, lo chiameremo Basic-micatanto.

Ad esempio la linea:

```
100 Persona = (Nome = "Mario", Cognome = "Rossi")
```

assegnerà alla variabile persona la coppia mostrata. Per selezionare i campi, è disponibile l'operatore OF:

```
110 PRINT Nome OF Persona
```


ListaPersone(1), restituirà:

("Mario", "Rossi")

ListaPersone(2) restituirà:

("Vito", "Bianchi")

Si può ad esempio scrivere:

130 A\$ = Nome OF ListaPersone(1)

A\$ conterrà "Mario".

Altro operatore per le sequenze, è ISIN. Serve per sapere se un oggetto appartiene o no ad una sequenza. La sua sintassi è:

Elemento ISIN Sequenza

Si usa negli IF, ad esempio:

140 IF 2 ISIN AltraLista THEN 200

salta a 200 se 2 appartiene a AltraLista (nel nostro caso si).

E per quanto riguarda i dati, ci fermiamo qui: vedremo ora come organizzare insieme di dati.

Le Classi

Il passo successivo è quello di organizzare per classi i dati che vogliamo trattare. Tornando all'esempio dato in apertura, quello di una biblioteca, le classi dati che prenderemo in considerazione saranno quelle dei materiali, dei materiali in prestito, di tutti gli utenti e delle prenotazioni. La caratteristica principale delle organizzazioni dei dati coi SGBD, è di poter in qualche modo legare dati di classi diverse. Ad esempio, ad ogni elemento della classe dei materiali prestati, può essere associato il corrispondente utente che ha in possesso il libro. Allo stesso modo, considerando la classe degli utenti, ad ogni elemento di quest'ultima saranno legati elementi della classe materiali in prestito. Per così dire, in una Base di Dati, i dati sono tra loro correlati: senza ovviamente ridondanza (duplicazioni di elementi). Se ad esempio Mario Rossi ha in prestito il Knuth vol. 1, nella classe Utenti, nella posizione "Mario Rossi", sarà presente un puntatore alla classe materiali prestati, precisamente all'elemento Knuth vol. 1.

Per visualizzare la descrizione di una Base di Dati, si utilizzano dei grafici. In figura 1 è visibile lo schema della biblioteca descritta. Le classi vengono rappresentate da rettangoli, con dentro il nome della classe. Fra le classi, gli archi interposti sono le associazioni tra dati. La freccia, singola o doppia, sta per associazione univoca o multipla: se è associato uno o più elementi della classe adiacente. Sempre riferendoci a questo schema, un utente della biblioteca può avere più libri in prestito, ma ogni libro prestato può stare al più presso un (e un solo) utente. Idem dicasi per le prenotazioni: una prenotazione riguarda un mate-



Figura 3 - L'installazione di una Base di Dati, per qualsiasi applicazione, coinvolge più fasi. Si parte dai Bisogni degli Utenti: ciò che i committenti della Base vogliono automatizzare. Il primo vero e proprio passo è la raccolta dei Requisiti: in questa fase si analizzano le procedure (manuali) già esistenti, estraendo da queste il comportamento che dovrà avere il sistema da realizzare. La Progettazione Concettuale serve per dare una definizione precisa della Base di Dati: è in questa fase che si partoriscono i vari schemi (fig. 1). A questo punto si è pronti per la realizzazione vera e propria: a partire dal linguaggio di programmazione specifico per Basi di Dati di cui si dispone, si traducono in programmi e procedure tutti gli schemi progettati al passo precedente.

riale della biblioteca, ma un libro può essere citato in più prenotazioni.

In alcuni sistemi sono presenti anche altri meccanismi per la manipolazione dei dati: le sottoclassi. Una sottoclasse è un sottoinsieme di una classe più grande detta Padre. Potremmo vedere la classe materiali prestati, come una sottoclasse della classe materiali. Col meccanismo delle sottoclassi è scongiurata anche la minima ridondanza dei dati: gli elementi della sottoclasse non sono copie degli elementi della classe Padre, ma semplicemente un modo diverso di vedere gli stessi oggetti. Supponiamo che ogni materiale, oltre a Titolo, Autore e Posizione nella biblioteca, abbia un campo denominato prestato che assume come valori "si" o "no" a seconda che sia in possesso di qualche utente o disponibile in biblioteca. Per definire la sottoclasse basta indicare che a Materiali Prestati appartengono tutti e solo i materiali con Prestato = "si". A questo punto, chiedersi se un libro appartiene alla sottoclasse materiali prestati equivale a controllare che Prestato sia uguale a "si".

Il tutto nella classe Padre, anche se l'interrogazione riguardava l'appartenenza alla classe figlia. Semplicemente un modo per rendere ancora più chiara la descrizione della Base di Dati, e minimizzare ancora di più la ridondanza.

Nelle prossime puntate

Nei prossimi numeri ci occuperemo un po' più di vicino della realizzazione di una base di dati. Il Basic-micatanto, con i suoi operatori per le Classi (li vedremo sul prossimo numero) si presta molto bene per descrivere e far funzionare Data Base col modello semantico dei dati: descrizione di questi secondo il loro vero significato nella realtà: un Utente è un utente, una persona... e come tale va trattata (non è un numero!).

Teniamo a precisare che il Basic-micatanto è sì inventato di sana pianta, ma non è molto dissimile dai sistemi oggi in commercio. Va molto bene per noi che siamo maledettamente abituati ai numeri linea e ai GOTO.

Daremo anche uno sguardo agli altri modelli di rappresentazione dei dati: il Modello Gerarchico, quello Reticolare e quello Relazionale. Quest'ultimo in particolare sta facendo parlare il mondo per la sua potenzialità unita ad una facilità d'uso da far spavento.

Infine presenteremo un Data Base per il Commodore 64 (strano!) scritto in Basic, ma opportunamente compilato per annullare i tempi morti di attesa. Segue anch'esso il modello semantico, con molti degli operatori del Basic-micatanto e un tocco di Relazionale, per essere al passo coi tempi. Il suo nome: Galileo/J. È tratto dal linguaggio Galileo per Basi di Dati, tuttora in fase di realizzazione presso l'Università degli Studi di Pisa, dipartimento di Scienze dell'Informazione.