

## **Capitolo quinto**

### **I RDBMS e gli Oggetti**

---

## 5.1.Introduzione

I DBMS di tipo relazionale puro mostrano sempre più una certa limitatezza rispetto a quelle che sono le nuove esigenze che si fanno avanti nelle aziende. I DBMS hanno forse risentito troppo dell'influenza delle aree in cui per primi trovarono applicazione, vale a dire quella gestionale ed amministrativa. La grande limitatezza deriva dalla possibilità di trattare solo dati di carattere strutturato come numeri e testi brevi, quindi dati di carattere semplice. Ma la maggior parte delle informazioni aziendali è in forma non strutturata. Appunti, rapporti, lettere, ed articoli di giornali sono solo alcuni esempi che mal si adattano alla struttura di un RDBMS puro. Il problema, ad esempio, con questo genere di materiali è che gli elementi d'interesse sono dispersi all'interno dei documenti e i DBMS sono organizzati solo per dati strutturati in cui i requisiti di ricerca sono stati definiti in anticipo. Inoltre, altre informazioni di carattere multimediale stanno acquistando sempre più peso nell'ambito delle aziende, ma non solo in queste<sup>1</sup>.

---

<sup>1</sup> "A chi serve un DBMS multimediale ?

Probabilmente non sono molte le realtà aziendali che possono avvantaggiarsi dall'aver un database completamente multimediale. Tra queste rientrano le aziende per le quali il prodotto aziendale "è multimediale" o è descritto prevalentemente in modo multimediale. (ad esempio società di distribuzione di audiovisivi, produttori cinematografici, gallerie d'arte e così via). La situazione cambia se invece pensiamo ad un database classico dove alle informazioni tipiche dell'ambiente gestionale (codice, descrizione, importo...) vengono aggiunte informazioni multimediali, come una foto del prodotto, un filmato che ne descrive le caratteristiche peculiari, un audiovisivo che descrive le modalità di impiego e così via.....questo DBMS può essere un valido strumento per il manager. Facciamo un esempio pratico, supponendo d'essere alla guida di una azienda che opera sia a livello nazionale che internazionale e vende prodotti per il mercato finale di livello medio-alto. Uno degli aspetti che potrebbe incidere sui fattori critici di successo sarebbe la possibilità di rilevare una correlazione statistica significativa tra l'andamento dei margini di contribuzione dei vari prodotti nelle varie aree geografiche o nei vari canali distributivi ed alcuni aspetti caratterizzanti, tipo di colore o la forma predominante di quegli articoli. Ad esempio, sarebbe molto interessante scoprire che l'incremento delle vendite sul mercato USA (cultura anglosassone) è fortemente correlato ad articoli, supponiamo, di colore blu e forme squadrate, mentre gli stessi colori e tipologia di forma ha avuto risultati esattamente contrari sul mercato brasiliano (cultura latino-americana), dove, invece, un incremento analogo si è verificato per colori predominanti sul rosso e forme tondeggianti. Ovviamente, questo tipo di correlazione senza la possibilità di elaborare la semantica delle informazioni

Cresce, di conseguenza, l'esigenza di gestire e al meglio i dati di carattere complesso, siano questi intere tabelle, testi, immagini, filmati ecc.. Questa esigenza ha spostato l'attenzione sugli oggetti, capaci di inglobare in se sia dati che operazioni che questi devono svolgere. Avremo, perciò, un oggetto tabella, uno testo, uno immagine ecc..

Si cerca sempre più da parte dei produttori di DBMS di avvicinare il mondo degli oggetti a quello relazionale, sia cercando di realizzare un vero e proprio "matrimonio" fra la struttura del modello relazionale con quello ad oggetti, sia realizzando delle interfacce ad oggetti in modo tale che il DBMS relazionale si presenti esternamente come un DBMS ad oggetti. Questa serie d'interfacce o più propriamente di strumenti, proprio perché utilizzano il comune concetto d'oggetto, risultano essere alla portata anche di una utenza non prettamente specializzata. Tale utenza, infatti, può così ricorrere ad un accesso ai dati aziendali senza l'ausilio di uno specialista.

## 5.2. Il modello orientato agli oggetti

Il modello orientato agli oggetti<sup>2</sup> è stato elaborato nell'ambito delle ricerche effettuate nel laboratorio di Palo Alto di Xerox in California negli anni '70 e nasce innanzi tutto come metodo di programmazione. Il modello orientato agli oggetti e la progettazione orientate agli oggetti introducono un modo diverso di realizzare software e di pensare le basi di dati che si avvicina di

---

multimediali non sarebbe possibile. E se questo semplice esempio lo immaginiamo calato nella realtà di una multinazionale, quale potrebbe essere una azienda produttrice di automobili, articoli di alta moda, si può immaginare quali scenari si possono dischiudere." M. Del Corto :Interrogazioni non strutturate, Zerouno 1/1995

<sup>2</sup> Non esiste un vero e proprio modello ad oggetti come può esserlo tanto per fare un esempio quello relazionale in campo database ma più propriamente si parla di "un insieme di tecniche e di idee che, riunite assieme realizzano un particolare modo di progettare software e basi di dati" S.Polini Introduzione alla programmazione orientata all'oggetto, Mcmicrocomputer 11/90.

In modo particolare per quanto riguarda le basi di dati "...le basi di dati hanno bisogno di un opportuno modello dei dati, e, malgrado l'assenza di uno standard, è possibile raggruppare alcuni concetti del paradigma ad oggetti comunemente accettati in un insieme detto core model o modello di base" E.Bertino - L.D. Martino :Sistemi di basi di dati orientate agli oggetti"

più alla realtà come la percepisce una persona comune<sup>3</sup>. L'elemento centrale di questo modello sono gli oggetti che raccolgono in una sola entità la struttura dei dati ed il loro comportamento. Questi oggetti possono "vivere di vita propria" potendo essere combinati in un programma. L'oggetto è una entità "schermata" che non può essere utilizzato in modo diverso da quello voluto da chi lo ha progettato e verrà a costituire oggetti più complessi combinandosi con altri oggetti. Infatti, con questo modello si realizza quella che viene chiamata la modularità del software e il riutilizzo del codice, cosa che nei linguaggi di tipo tradizionale non veniva permesso, in quanto un programma tradizionale può essere visto come un insieme di blocchi annidati in cui dati e procedure sono separati<sup>4</sup>.

L'oggetto può definirsi come *una entità astratta che gode di certe proprietà che può essere manipolata in modi diversi definiti a priori*<sup>5</sup>. I programmi eseguono i loro calcoli trasferendo messaggi fra gli oggetti attivi, che corrispondono, nel computer, alle entità nel mondo reale<sup>6</sup>.

Concetti significativi all'interno di questo modello sono quelli di

- ? *astrazione*
- ? *oggetto*
- ? *incapsulamento*
- ? *classe*
- ? *ereditarietà*
- ? *polimorfismo*.

---

<sup>3</sup> Questo modo di concepire i software conduce sia ad un più diretto coinvolgimento dell'utente alla fase di progettazione del software stesso poiché i problemi si riconducono a concetti che sono di più facile comprensione per lo stesso sia a ad un più facile ed immediato utilizzo del software. Mahesh, Dodani, Hughes, Moshell : La programmazione OO parte 3, Byte 9/1989

<sup>4</sup> "Nei sistemi di programmazione tradizionali, dati e procedure sono entità separate ; il programmatore ha la responsabilità di applicare procedure attive a strutture dati passive e, spesso, di accertare che una procedura lavori correttamente con i tipi di dati a cui è applicata. Al contrario un sistema di programmazione orientata ali oggetti non vede un oggetto semplicemente come un dato passivo, ma come una combinazione del suo stato privato e dei metodi che lo manipolano." D. Thomas :La programmazione OO parte 1 Byte 6/1989

<sup>5</sup> C. Giustozzi Cos'è la OOP Mcmicrocomputer 11/1990

<sup>6</sup> D. Thomas :La programmazione OO parte 1 Byte 6/1989.

Tutti questi concetti fondendosi insieme danno luogo ad un nuovo modo di concepire la realizzazione del software e delle basi di dati che ha come vantaggi quello di permettere una realizzazione modulare del software tramite la composizione d'oggetti complessi a partire da oggetti più semplici "discreti" (cioè che hanno vita propria) e di realizzare una migliore riutilizzazione del codice. Tutto questo ha portato, ad esempio, alla realizzazione d'ambienti caratterizzati da una grafica molto sviluppata che ha permesso di realizzare un approccio più semplificato alle risorse software ed hardware proprio per la presenza, anche visiva, d'oggetti che sintetizzano in maniera più naturale i concetti, come l'uomo li percepisce dalla realtà. Esempio lampante di questo nuovo modello è il grande successo che hanno avuto le varie versioni di Windows e tutti i prodotti ad esse legati, basati sulla realizzazione di un ambiente grafico estremamente sviluppato che fanno degli oggetti l'elemento cardine del loro funzionamento.

### **5.2.1. L'astrazione**

L'astrazione è il concetto "principe" di tutti i modelli compreso quello ad oggetti. Dell'astrazione abbiamo discusso diffusamente nel capitolo dedicato alla progettazione concettuale affermando che si tratta di un processo mentale attraverso il quale si evidenziano alcune proprietà e caratteristiche in un insieme d'oggetti che si ritiene importante escludendone altre che sono ritenute irrilevanti arrivando, poi, a definire un nuovo oggetto che contiene in se tutte le proprietà considerate. Quindi, tramite l'astrazione si determinano le caratteristiche di un oggetto e la sua aggregazione in classi che, nel modello ad oggetti, hanno carattere gerarchico.

### 5.2.2. Oggetti ed incapsulamento

Uno degli elementi fondamentali del modello ad oggetti è quello che riguarda la definizione di un oggetto. Questi oggetti sono costruiti per avvicinarsi il più possibile a quella che è la concezione umana d'oggetto e, quindi, per rendere i sistemi informatici più adatti a rappresentare il mondo reale.

Negli oggetti che incontriamo tutti i giorni, non facciamo mai distinzione fra gli elementi fisici di cui l'oggetto è costituito ed il loro comportamento<sup>7</sup>. Elementi essenziali per la composizione di un oggetto sono i dati e i metodi. I dati possono essere variabili o costanti o in generale informazioni, mentre i metodi sono rappresentati dalle procedure, funzioni di manipolazione e in generale operazioni che si possono compiere su di esso. Nella programmazione tradizionale le strutture dei dati, che dovranno contenere le informazioni, saranno inserite da una parte mentre le procedure e le funzioni di manipolazione dall'altra. I programmi OO non vengono organizzati in procedure che condivideranno dei dati "globali", cioè accorpati tutti insieme. Al contrario *i dati vengono incapsulati con le procedure che possono accedervi. Questo concetto è spesso chiamato astrazione dei dati o programmazione modulare*<sup>8</sup>.

Nome :Maurizio Cognome :Boghetto Via :Lavagnini n.23 matricola 14545..... .....	<b>Dati dello oggetto</b>
Cammina Mangia dorme studia .....	<b>Metodi (o operazioni) Dello Oggetto</b>

**Figura 5.1 Esempio di rappresentazione di un oggetto (istanza di una classe)**

<sup>7</sup> Si può pensare l'oggetto come un "tipo di dato astratto". "Per tipo di dato astratto si intende, un tipo di dato che non venga caratterizzato semplicemente da valori che possono assumere variabili ad esso appartenenti ma anche da un insieme di operazioni". Questo concetto espresso da Polini trova applicazione nell'inserire in strutture relazionali tipi di dati che non siano semplicemente alfanumerici ma anche dati di carattere più complesso come quelli, ad esempio di carattere multimediale.

<sup>8</sup> D. Thomas, La programmazione OO - Bit 6/1989