

DB III contro tutti

seconda parte

Completiamo, con questo secondo articolo, l'analisi delle motivazioni del successo del Data Base III, della Ashton Tate, da anni best seller incontrastato nella categoria prodotti DBMS (gestione di banche dati) su PC. Lo facciamo parlando non solo del DB III ma anche dei prodotti concorrenti, che in questi ultimi mesi si stanno facendo più... concorrenziali, sia in termini di caratteristiche tecniche, che in termini di aggressività sul mercato. E questo in un settore merceologico, quello dell'Informatica Individuale, che, essendo diventato «di massa», adotta le stesse regole del gioco proprie di tale tipo di mercato

Nello scorso articolo abbiamo appunto detto che il successo del DB III è dipeso non solo dalle proprie qualità intrinseche, ma anche da una accorta politica di diffusione e di mercato, che lo ha visto uscire sempre nel momento più opportuno, sin dai tempi delle macchine 8 bit, quando ancora si chiamava DB II, sia alla nascita del PC IBM.

Ricordiamo anche che per DBMS, Data Base Management System, intendiamo, non il semplice Filer, adatto alla gestione di un solo archivio, ma lo strumento con il quale gestire un insieme di archivi variamente organizzati e variamente correlati tra di loro.

Poiché il DBMS aiuta a risolvere problemi complessi è sempre necessaria una preventiva comprensione del problema stesso. È molto utile ad esempio una sua schematizzazione, che prescindendo sia dal prodotto che si utilizzerà sia, addirittura, dall'uso di un computer, e che permetta di individuare i vari archivi e il loro contenuto.

Ricordiamo che su questa materia, fondamentale per l'Informatica, esistono centinaia di libri il cui scopo è quello di proporre dei linguaggi e dei simbolismi grafici che permettono di produrre, come risultato dell'analisi, uno schema grafico di facile ed universale comprensione.

Fatta questa premessa sulle metodolo-

gie di soluzione di un problema complesso torniamo ai nostri prodotti software.

Nel corso di questi due articoli sull'argomento DBMS, citiamo e proviamo alcuni prodotti, già noti ai nostri lettori, che pur avendo una finalità identica al DB III, sono operativamente e filosoficamente differenti.

Per entrare concretamente nel problema prendiamo in esame un semplice caso comprendente tre archivi, e iniziamo subito con il formalizzare il problema.

Il lavoro con più archivi Caso di studio

Una industria produce articoli di cancelleria (Archivio Articoli) e li rende ai propri clienti (Archivio Clienti). La vendita avviene attraverso degli ordini (Archivio Ordini).

I campi presenti nei tre archivi sono chiaramente indicati nello schema di figura 1 e nella stampa, eseguita da DB III, delle strutture e del contenuto degli archivi di figura 2.

Tra i tre archivi esistono inoltre due evidenti legami, il codice del cliente dell'archivio ordini è in relazione con il codice cliente dell'archivio clienti, e il codice dell'articolo dell'archivio ordini è in relazione con il codice articolo presente nell'archivio articoli. E anche questo è indicato nello schema di figura 1.

Più concretamente, nell'utilizzare l'archivio ordini, ad esempio nel valorizzare un singolo ordine è necessario avere contemporaneamente disponibili tre record dei tre archivi, e quindi contemporaneamente disponibili i campi dei tre record. Occorre avere per ciascun ordine anche i dati relativi al cliente che ha eseguito l'ordine e relativi all'articolo che è stato ordinato.

Evidentemente il caso proposto comporta una semplificazione inaccettabile in una applicazione reale, in quanto un archivio ordini è sempre composto da due parti, una testata in cui sono memorizzati i dati generali dell'ordine e da più righe d'ordine, ciascuna delle quali riguarda un singolo articolo.

Qualsiasi prodotto di tipo DBMS è in grado di riconoscere i legami tra i vari archivi e di occuparsi sia di individuare i record collegati per mezzo del legame, sia di renderne disponibili i dati per le necessarie elaborazioni.

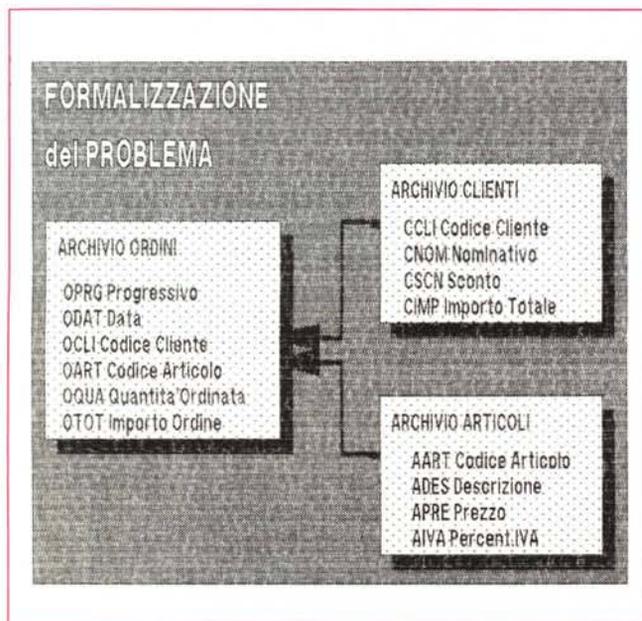


Figura 1
Schematizzazione del caso studio.
L'utilizzo efficace di prodotti DBMS per la gestione di basi di dati (intese come insieme di archivi tra loro correlati) richiede una preventiva analisi a schematizzazione del problema. In questa fase iniziale vanno individuate le entità in gioco e le relazioni tra di esse.

```
Struttura del file : A:CLIENTI.dbf
Numero totale record : 3
Data ultima revisione: 30/03/89
Campo Nome campo Tipo Dim Dec
1 CCLI Carattere 4
2 CNOM Carattere 14
3 CSCN Numerico 5 2
4 CIMP Numerico 9
Totale: 33
```

```
Record CCLI CNOM CSCN CIMP
1 A001 ROSSI 15.00
2 A002 VERDI 20.00
3 A003 BIANCHI 18.00
```

```
Struttura del file : A:ARTICOLI.dbf
Numero totale record : 3
Data ultima revisione: 30/03/89
Campo Nome campo Tipo Dim Dec
1 AART Carattere 5
2 ADES Carattere 20
3 APRE Numerico 6
4 AIVA Numerico 2
Totale: 34
```

```
Record AART ADES APRE AIVA
1 00001 QUADERNO 500 18
2 00002 CARTELLA 23750 18
3 00003 PENNARELLI 1700 18
```

```
Struttura del file : A:ORDINI.dbf
Numero totale record : 10
Data ultima revisione: 30/03/89
Campo Nome campo Tipo Dim Dec
1 OPRG Carattere 4
2 ODAT Data 8
3 OCLI Carattere 4
4 OART Carattere 5
5 OQUA Numerico 4
6 OTOT Numerico 12
Totale: 38
```

```
Record OPRG ODAT OCLI OART OQUA OTOT
1 0001 02/01/87 A001 00002 4 0
2 0002 03/02/87 A003 00003 10 0
3 0003 03/03/87 A001 00001 50 0
4 0004 09/04/87 A003 00002 12 0
5 0005 11/04/87 A002 00001 100 0
6 0006 12/05/87 A002 00003 20 0
7 0007 21/05/87 A002 00002 10 0
8 0008 17/06/87 A001 00003 35 0
9 0009 21/06/87 A003 00002 5 0
10 0010 04/07/87 A002 00001 100 0
```

Figura 2 - DB III
Gli archivi utilizzati.
Nel nostro caso studio utilizziamo, con i vari prodotti in esame, gli stessi archivi (Clienti, Articoli e Ordini), con gli stessi nomi dei campi caricati con gli stessi dati. In questo caso vediamo le loro strutture in DB III e i loro contenuti visualizzati con il comando LIST.

su un campo univoco) oppure per organizzare i dati (ad esempio se l'indice è costruito su un campo di raggruppamento).

La seconda funzionalità fondamentale è costituita dalle Relazioni, con le quali vengono realizzati dei collegamenti tra gli archivi.

Anche la relazione non è strutturale. Anzi va semplicemente dichiarata senza che il DB III esegua un controllo formale della sua correttezza. Se non è corretta, così come se tra gli archivi non esistono corrispondenze, si ha come conseguenza quella di non aver... risultati. La Relazione non dà mai messaggi di errore.

Per dichiarare una relazione tra due archivi occorre aprirli entrambi, ovviamente su aree differenti, il secondo deve essere indicizzato sul campo in comune, e la relazione va dichiarata, dalla prima area indicando il campo del primo archivio che è in relazione con il secondo archivio:

```
SET RELATION TO CAMPO1 INTO ARCHIVIO2
```

Sottointendendo che il secondo archivio è indicizzato su un campo, di qualsiasi nome, che è corrispondente al CAMPO1 del primo archivio.

A questo punto a ciascun record del primo archivio è collegato il record del secondo cui corrisponde il campo relazionato.

Sono quindi disponibili i campi dei due record, i primi richiamabili con il loro nome, e i secondi richiamabili facendo precedere il loro nome da una lettera che indica l'ordine di apertura dell'archivio e da un simbolo costruito con il segno meno e il segno maggiore (es. C->campo).

Occorre fare attenzione al fatto che la relazione cerca sul secondo archivio il campo comune, e quindi ne trova solo uno. Se ce ne sono altri non vengono trovati.

Quindi la relazione usata nel modo citato vale solo nei casi in cui ci sia corrispondenza 1 a 1, oppure N a 1 tra il primo e il secondo archivio.

Nella figura 3 vediamo una serie di comandi DB III che eseguono quattro elenchi.

Il risultato è in figura 4.

Il primo elenco contiene dati dall'archivio ordini. Il secondo si avvale della relazione con l'archivio clienti. Il terzo della relazione con l'archivio articoli. Que-

In altre parole occorre aprire gli archivi e dichiarare le relazioni, dopodiché è il prodotto DBMS che provvede a prelevare i dati là dove risiedono. È questo il concetto di relazionalità tra gli archivi.

Prima di passare alla fase pratica premettiamo che la finalità dell'articolo non è ovviamente quella di permettervi di costruire una procedura, ma è quella di sperimentare, con vari prodotti e quindi con vari metodi operativi, un lavoro eseguito su più archivi relazionati tra di loro.

Affrontiamo il caso con il DB III

Eseguita l'analisi, che come detto, prescinde dal prodotto che si intende utilizzare, si passa a costruire le strutture degli archivi.

Successivamente, per sperimentare la correttezza delle operazioni che si eseguiranno, è bene inserire «a manina» un po' di dati, funzionali alle prove che si vorranno fare e quindi con codici clienti e codici articoli uguali nei vari archivi.

Nel DB III esistono due funzionalità fondamentali, la cui padronanza è indispensabile, per il suo corretto utilizzo e per il reale sfruttamento delle sue enormi potenzialità.

La prima funzionalità è costituita dall'Indice, che assume dignità di archivio e che permette di «vedere» nell'ordine logico voluto l'archivio, indipendentemente dal reale ordine fisico che questo ha.

L'indice non è un elemento strutturale, che va definito al momento della creazione dell'archivio, ma può essere costruito anche estemporaneamente all'occorrenza. Altra sua caratteristica è quella di non essere necessariamente legato ad un campo dell'archivio, ma di poter essere legato a qualsiasi espressione sintatticamente valida (ad. es. una sottostringa di un campo, un campo di un archivio esterno relazionato, ecc.).

L'Indice ha due utilizzazioni differenti. Serve per creare delle chiavi di ricerca rapida (ad esempio se l'indice è costruito

sto perché non è possibile dichiarare due relazioni da uno stesso archivio verso due altri archivi. Nella parte finale viene prodotta una stampa in cui sono convogliati dati dai tre archivi. Viene utilizzato un comando di programmazione (non utilizzabile in comandi diretti)

DO WHILE .NOT. EOF () skip ENDDO che produce uno scorrimento completo dell'archivio.

Per ciascun record viene dichiarata la relazione prima verso l'archivio clienti e poi verso l'archivio articoli. In tal modo vengono prelevati i dati là dove risiedono. Ad esempio la valorizzazione è ottenuta mediante un calcolo che comprende il dato quantità (archivio ordini), prezzo e IVA (archivio articoli) e sconto (archivio clienti).

Va detto che la dichiarazione di una relazione è un'operazione che non richiede apertura/chiusura di file, e quindi è una operazione che non rallenta l'esecuzione dell'elaborazione.

Figura 3 - DB III
Programma che utilizza i tre archivi. In DBIII sia gli INDICI, che permettono di vedere nel voluto ordine logico un certo archivio, sia le RELAZIONI, che mettono in corrispondenza record di più archivi secondo uno o più campi in comune, non sono elementi strutturali (che vanno quindi definiti al momento della costruzione della struttura) ma possono essere anche creati in maniera estemporanea quando occorre.

```
* ARTICOLO
SET TALK OFF
SET SAFETY OFF
SELECT 3
USE CLIENTI
INDEX ON CCLI TO CLIENTI
SELECT 2
USE ARTICOLI
INDEX ON AART TO ARTICOLI
SELECT 1
USE ORDINI
INDEX ON OPRG TO ORDINI
set print on
LIST ODAT,OCLI,OART,OQUA
SET RELATION TO OCLI INTO CLIENTI
LIST ODAT,OCLI,C->CNOM,C->CSCN,OART
SET RELATION TO OART INTO ARTICOLI
LIST ODAT,OCLI,OART,B->ADES,B->APRE,B->AIVA
GO TOP
TOT=0
DO WHILE .NOT. EOF()
  SET RELATION TO OCLI INTO C
  VNOM=C->CNOM
  VSCN=(1-C->CSCN/100)
  SET RELA TO OART INTO ARTICOLI
  VDES=B->ADES
  VPRE=B->APRE
  VIVA=(1+B->AIVA/100)
  VIMP=OQUA*VPRE*VIVA*VSCN
  TOT=TOT+VIMP
  ? OPRG,ODAT,VNOM,OART,VDES,OQUA,VIMP
  SKIP
ENDDO
? SPACE(60),TOT
CLOSE DATABASES
```

Record	ODAT	OCLI	OART	OQUA
1	02/01/87	A001	00002	4
2	03/02/87	A003	00003	10
3	03/03/87	A001	00001	50
4	09/04/87	A003	00002	12
5	11/04/87	A002	00001	100
6	12/05/87	A002	00003	20
7	21/05/87	A002	00002	10
8	17/06/87	A001	00003	35
9	21/06/87	A003	00002	5
10	04/07/87	A002	00001	100

Record	ODAT	OCLI	C->CNOM	C->CSCN	OART
1	02/01/87	A001	ROSSI	15.00	00002
2	03/02/87	A003	BIANCHI	18.00	00003
3	03/03/87	A001	ROSSI	15.00	00001
4	09/04/87	A003	BIANCHI	18.00	00002
5	11/04/87	A002	VERDI	20.00	00001
6	12/05/87	A002	VERDI	20.00	00003
7	21/05/87	A002	VERDI	20.00	00002
8	17/06/87	A001	ROSSI	15.00	00003
9	21/06/87	A003	BIANCHI	18.00	00002
10	04/07/87	A002	VERDI	20.00	00001

Record	ODAT	OCLI	OART	B->ADES	B->APRE	B->AIVA
1	02/01/87	A001	00002	CARTELLA	23750	18
2	03/02/87	A003	00003	PENNARELLI	1700	18
3	03/03/87	A001	00001	QUADERNO	500	18
4	09/04/87	A003	00002	CARTELLA	23750	18
5	11/04/87	A002	00001	QUADERNO	500	18
6	12/05/87	A002	00003	PENNARELLI	1700	18
7	21/05/87	A002	00002	CARTELLA	23750	18
8	17/06/87	A001	00003	PENNARELLI	1700	18
9	21/06/87	A003	00002	CARTELLA	23750	18
10	04/07/87	A002	00001	QUADERNO	500	18

0001	02/01/87	ROSSI	00002	CARTELLA	4	95285.0000
0002	03/02/87	BIANCHI	00003	PENNARELLI	10	16449.2000
0003	03/03/87	ROSSI	00001	QUADERNO	50	25075.0000
0004	09/04/87	BIANCHI	00002	CARTELLA	12	275766.0000
0005	11/04/87	VERDI	00001	QUADERNO	100	47200.0000
0006	12/05/87	VERDI	00003	PENNARELLI	20	32096.0000
0007	21/05/87	VERDI	00002	CARTELLA	10	224200.0000
0008	17/06/87	ROSSI	00003	PENNARELLI	35	59678.5000
0009	21/06/87	BIANCHI	00002	CARTELLA	5	114902.5000
0010	04/07/87	VERDI	00001	QUADERNO	100	47200.0000
						937852.2000

Figura 4 - DB III - Output del programma. Sono quattro stampe dall'archivio Ordini. Nella prima solo dati di questo archivio, nella seconda e nella terza dati anche dall'archivio clienti e dall'archivio articoli. Nell'ultima stampa, eseguita con l'istruzione di programmazione DO WHILE .. END, appaiono dati da tutti e tre gli archivi.

L'approccio Data Ease

Il Data Ease è un prodotto più recente, rispetto al Data Base III, e, contrariamente a quest'ultimo, trasferisce al momento iniziale della costruzione della struttura dell'archivio la definizione sia di quali siano i campi indice (possono solo essere campi) sia di quali siano i legami con gli altri archivi.

Questo comporta il fatto che anche la relazione tra due archivi, che avviene in generale attraverso uno o più campi in comune, diventa un elemento strutturale.

Elenchiamo qui di seguito in maniera schematica i principali passi operativi necessari per risolvere, con il Data Ease, il nostro caso studio.

1 Schematizzazione del problema con impostazione del nome dei campi di ciascuna struttura e con simbolismo che identifichi le relazioni tra i vari elementi (fig. 1);

2 Creazione dei vari moduli (maschera più struttura sottostante) rimandando ad un momento successivo l'inserimento dei campi calcolati e dipendenti dagli altri archivi (figg. 5 e 6);

3 Definizione, per mezzo dell'apposita procedura, delle relazioni tra i vari archivi (fig. 7);

4 Completamento della creazione dei moduli, inserendo anche i campi dipendenti dalle relazioni, sia di semplice inquiry sia di calcolo vero e proprio;

5 Creazione di uno o più report.

Questo termine in Data Ease che indica sia il report vero e proprio sia una

ORDINI

REGISTRAZIONE ORDINI

PROGRESSIVO : 0/0001	DATA : 01/02/87
CODICE CLIENTE : C-0001	NOMINATIVO : MONTI
	SCONTO : 15,00%
CODICE ARTICOLI : A-00002	DESCRIZIONE : CARTOLINA
	PREZZO : 23,750 IVA : 10%
QUANTITA' : 4	TOTALE : 95,285

F2CONF F3VEDI F4FINE F5CANO MOD F6CANO CAMPO F7CANO REC F8MODIF F9REPORT F10EOL

elaborazione di tipo Batch (fig. 8);
6 Completamento della procedura con la creazione di un menu (anche questa funzionalità specifica del Data Ease) di accesso alle varie sottoprocedure di gestione maschere e di esecuzione report.
L'approccio offerto del Data Ease è quindi obbligato. Tutto quello che succede, dalla costruzione di una struttura alla interrogazione estemporanea di un archivio, realizzabile attraverso un Report rapido, avviene in un ambiente maschera.

Figura 6 - Data Ease.
La creazione della struttura di un archivio Data Ease si esegue in modalità full-screen.
In corrispondenza della zona destinata ad un campo si preme il tasto F10 e appare una maschera nella quale vanno interattivamente inserite tutte le caratteristiche strutturali relative a quel campo.

relazioni

MODULO RELAZIONE

TRA: Modulo 1 ORDINI e Modulo 2 ARTICOLI

BASETA SULL'UCORDIANZA TRA I SEGUENTI CAMPI (definire almeno una coppia di campi):

Campo OIRT = Campo ART

e Campo = Campo

e Campo = Campo

NOMI (OPZIONALI) NELLA RELAZIONE (I nomi dei moduli verranno usati per default)

Modulo 1: R2-ORD Modulo 2: R2-ART

F2CONF F3VEDI F4FINE F5CANO MOD F6CANO CAMPO F7CANO REC F8MODIF F9REPORT F10EOL

Ad una prima impressione, questa soluzione, anche se tendente alla massima facilità operativa, sembrerebbe troppo rigida per poter risolvere la larga gamma di problemi che si presentano in un'applicazione che lavora su più archivi.
In realtà le funzionalità offerte sono tutte quelle più comunemente necessarie e quindi il prodotto risulta idoneo alla soluzione standard di problematiche, anche su più archivi, standard.

Figura 8
Data Ease.
Creazione di un Report.
Anche la costruzione di un report avviene in maniera interattiva in una speciale maschera.
Si possono utilizzare tutte le relazioni definite in struttura, oppure, ed è il caso mostrato, creare al momento delle relazioni che vengono usate solo dal report.

Figura 5 - Data Ease.
Archivio ordini.
In questa maschera dell'archivio ordini appaiono anche i cosiddetti campi relazionati, che non sono presenti fisicamente nell'archivio ordini, ma vengono prelevati dai rispettivi archivi solo al momento della visualizzazione.

Paradox 2

Il Paradox 2 della Ansa/Borland lavora tramite un menu ad albero 123-like, che è una modalità operativa consolidata e di facile uso. In tal modo infatti le operazioni vengono molto semplificate in quanto il menu propone solo le funzioni possibili in quel dato momento logico.

MODULO ORDINI

Nome del campo : OIRT Numero
Tipo di campo : Intero
Tipo di numero : 9
Massimo numero di cifre nel campo :
Per saltare le restanti domande, premere CONFERMA, CANCELLA o MODIFICA
E' RICHIESTO l' inserimento di un valore nel campo? : no
Il campo richiede un accesso veloce (INDEXED)? : no
Il valore del campo dev' essere unico? : no
Il campo richiede un controllo di INTERVALLO di valori? : no
Valore DERIVATO (calcolo/lookup/sequenza/default)? : si
Formula associata al campo: OIRA * LOOKUP "R2-ART" ARTA * (100 - LOOKUP "R1-CLI" CSCH) / 100 * (1 + LOOKUP "R2-ART" ARTA / 100)
Occorre IMPEDIRE l' inserimento di dati? : no
Livello di sicurezza per LEGGERE il campo:
Livello di sicurezza per SCRIVERE nel campo:
Messaggio di AIUTO:
Attributi di visualizzazione del campo: Evid, 1

F2 CONFERMA F7 CANCELLA F8 MODIFICA

Figura 7 - Data Ease.
Anche la relazione tra gli archivi viene definita in una maschera molto chiarificatrice, in cui fisicamente vanno accostati i nomi dei campi comuni a due archivi. Ringrazio per la preziosa consulenza l'amico Luigi Sandulli, specialista a livello professionale del Data Ease.

Inoltre poiché esiste sempre una funzionalità Do-It (esegui) e una Cancel (non esegui) non si corrono grossi rischi di combinare dei guai neanche se si dovessero sbagliare delle scelte di menu.

Nel nostro caso, aprendo i tre archivi di prova, li troviamo tutti e tre direttamente visualizzati, nella classica forma tabellare, contemporaneamente sul video. Si passa da uno all'altro muovendosi con tasti funzione, e all'interno di

STAMPA ORDINI R I C 1

```
for ORDINI
list records
OPRG :
ORDI in order ;
any R1-CLI CNOH in order ;
OIRT ;
any R2-ART ARTA ;
OIRA ;
OIRA * any R2-ART ARTA * ( 100 - any R1-CLI CSCH ) / 100 *
( 1 + any R2-ART ARTA / 100 ) : item sum .
```

F1INTERATTI F2CONF F3TAGLIA F4FINE F5SCOPIA F6INCOLLA F7CANO LIN F8INS LIN F9LIV

sce un vero e proprio mondo a se stante.

Parliamo del concetto di Macro che in «dibitrees» significa la possibilità di attribuire ad una variabile il valore di comando, o di parte di esso, o di formula e di ... insomma di qualsiasi cosa. Per far assumere alla variabile il suo reale contenuto occorre farla precedere da un semplice e discreto segno «&» (altrimenti detto E commerciale) e il DB III ne interpreterà il contenuto.

Se il contenuto è il comando eseguirà il comando, se il contenuto è un'operazione matematica eseguirà il relativo calcolo, e così via. In figura 12 proponiamo degli esempi di Macro con le relative spiegazioni. Si lavora con un archivio Clienti che comprende i campi Gruppo, Società e Provincia.

La Macro in DB III introduce il concetto di esecuzione del contenuto di una variabile.

Ma si può arrivare anche a ipotizzare più livelli, come ad esempio il contenuto del contenuto del contenuto... come in un gioco di scatole cinesi.

Si veda, nella figura, l'esempio in cui ad una variabile viene assegnato il valore di una formula matematica. Questo significa ad esempio che in una procedura DB III si può realizzare un program-

ma di calcolo in cui non esistono formule di calcolo, ma solo riferimenti a variabili, gestibili molto facilmente dal di fuori del programma. Anche se cambiano le regole del calcolo il programma non deve essere manomesso, basta cambiare il contenuto di una variabile.

Altro esempio proposto è quello di far assumere ad una variabile il nome di un campo (istruzione FIELD (X)) e di visualizzare non il contenuto della variabile (che è il nome del campo), ma il contenuto del contenuto... che è il dato.

In particolare questa serie di istruzioni è indipendente dall'archivio su cui lavora, nel senso che vale sempre in quanto riconosce da sola la struttura dell'archivio aperto al momento.

Conclusioni

Due considerazioni preventive.

La prima è che mentre in altre categorie di software (ad esempio Word Processor o Spreadsheet) si è arrivati ad un linguaggio concettuale universale, che prescindendo dal prodotto, nel mondo del DBMS le soluzioni proposte dai vari pacchetti possono essere assolutamente differenti le une dalle altre. E gli esempi mostrati lo evidenziano.

La seconda è che in ogni caso, e

quindi indipendentemente dal prodotto che si utilizza, il preventivo lavoro di schematizzazione del problema non viene mai rimesso in discussione. E questo a dimostrazione della sua validità intrinseca.

Al contrario con lo strumento spreadsheet il lavoro preventivo, a tavolino, di schematizzazione del problema, può essere superfluo, in quanto è lo spreadsheet stesso (da buon foglio di carta a quadretti) che può essere usato come strumento per studiare la schematizzazione.

Il Data Base III è orientato al comando. L'utente deve scrivere l'istruzione che viene direttamente eseguita. Se l'operazione da compiere è complessa occorreranno più comandi messi in sequenza.

Data Ease è orientato alla maschera. Qualsiasi operazione sia di interazione con l'archivio, sia di definizione di struttura, report, relazione, ecc. avviene attraverso una maschera.

In ciascuna maschera le operazioni possibili sono sempre legate all'uso dei tasti funzione, che conservano comunque, in qualsiasi situazione, il loro valore.

Con Paradox 2 infine si lavora con l'albero dei menu (che ormai si definisce 123-like). Ci si muove tra i rami e trovata la funzione la si attiva.

L'interazione con gli archivi avviene attraverso dei tracciati record che appaiono orizzontalmente sul video e nei quali vanno inserite le specifiche dell'operazione.

Quindi la conclusione più evidente è che nei prodotti DBMS non si è affermata una metodologia standard di lavoro, ogni prodotto ne adotta una che presenta vantaggi e svantaggi.

Ma va anche detto che in una procedura complessa che richieda l'uso di più archivi e di relazioni tra di essi, la difficoltà principale è sicuramente costituita dalla schematizzazione del problema, ovvero dello studio, che va fatto a tavolino, ancor prima di accendere il computer, di quali siano gli archivi, i campi, i campi chiave, le relazioni tra di essi.

Una volta chiarito e formalizzato il problema e... acceso il computer, le modalità operative con le quali comunicare con lo strumento DBMS assumono una importanza minore.

In altre parole le differenti modalità operative possono risultare rilevanti in una fase iniziale di apprendimento del prodotto, ma in seguito con il raggiungimento della padronanza e anche col raggiungimento di certi automatismi queste diventano del tutto secondarie.

```
* USO DELLE MACRO &

Z ="X*Y"
X =10
Y = 5
? Z,&Z
* esegue l'oper.ne contenuta in Z

A="USE CLIENTI"
&A
* macro=comando

B="CLIENTI"
USE &B
* macro=parte di comando

C="LIST GRUPPO,SOCIETA"
D="FOR PROVINCIA='MI'"
&C &D
* composizione di due macro

E="SOCIETA"
LIST E,&E
* macro di un campo

F=FIELD(3)
LIST &F
* funzione FIELD da il nome del campo

PIPPO="PLUTO"
* definizione variabile PIPPO="PLUTO"

&PIPPO="KATIA"
* def.ne var.le PLUTO (contenuto di PIPPO) ="KATIA"

N=41
V="V"+STR(N,2)
&V="PAPERINO"
* def.ne di una var.le (V41) contenuta in un'altra (V)
```

Figura 12 - Data Base III. Il fascino delle macro. In un uso avanzato del DB III, che oggi, è bene ricordarlo, è sicuramente il linguaggio di programmazione più diffuso per lo sviluppo di applicazioni su PC, risulta fondamentale l'uso delle MACRO. Con questa un comando o parte di esso può essere gestito tramite una comune variabile, che è facilmente manipolabile, e che al momento opportuno, può essere eseguita.