



Il DB III è un DBMS (Data Base Management System) scritto dalla Ashton Tate (la casa del DB II e del Framework) per le macchine 16 bit della categoria IBM e IBM compatibili.

Un DBMS è uno strumento software particolare, specializzato per la soluzione di problemi di organizzazione e manipolazione dati. Ha quindi un campo di applicazione ben più ristretto di un linguaggio «general purpose» (quale è ad esempio il Basic), o di un Package Integrato (quali il Framework o il Symphony).

Non ha nessuna attitudine non solo per i problemi scientifici, per i quali esistono linguaggi specializzati, ma in generale non permette neanche l'utilizzazione di tutte le caratteristiche hardware della macchina, e quindi non permette grafica, non permette suoni, non è in grado di gestire direttamente, con propri comandi, porte di comunicazione.

La sua sola funzione è svolgere bene la sua funzione specifica di manipolatore di dati, molto meglio ad esempio del Basic. E poiché si può dire che in tutte le procedure gestionali il problema principale è proprio quello della manipolazione dei dati, si spiega sia l'importanza dei DBMS nella

produzione di software applicativo, sia d'altro canto, lo scarso interesse che per gli stessi hanno gli hobbisti, più interessati ad usi «divertenti» dei propri computer.

Tra i DBMS per microcomputer quello che ha avuto il maggior successo è senza dubbio il DB II, prodotto dalla Ashton Tate alcuni anni fa per le macchine 8 bit, basate sul microprocessore Z80. Su tale Package la Ashton Tate, nata nel 1980, ha fondato le sue fortune (oltre 30 milioni di dollari di fatturato nel 1984).

Il DB III è strettamente legato al DB II, in quanto, anche se è restrittivo definirlo una sua nuova release, ne conserva del tutto la filosofia, e buona parte della

sintassi. È stato però totalmente riscritto (in linguaggio C) in modo da allineare le sue prestazioni a quelle permesse dalla nuova generazione di macchine. È quindi, rispetto al DB II, estremamente più potente.

Molti critici si sono chiesti il perché del successo del DB II, che a parere degli stessi critici è un prodotto abbastanza ostico per avere una diffusione di massa, e non privo di «talloni d'Achille», tipo il famigerato SORT. In realtà il DB II è uscito nel momento giusto, quello in cui le macchine 8 bit avevano raggiunto la massima potenza possibile in termini di memoria centrale e cominciavano a vedersi i primi hard disk. Le macchine, dal punto di vista hardware,

## Ashton Tate dBASE III

di Francesco Petroni



quindi permettevano un uso produttivo anche nel campo gestionale.

Ma il campo gestionale richiede procedure basate sulla manipolazione di grossi volumi di dati, e il Basic e i DOS elementari allora esistenti, non erano molto adatti a risolvere queste problematiche. Malgrado ciò la quasi totalità dei prodotti di software gestionale erano scritti in Basic.

E così l'arrivo di un DBMS serio e soprattutto programmabile, fu accolto con un entusiasmo che oggi può apparire eccessivo, da chi aveva cominciato a trovare il Basic insufficiente per risolvere complessi problemi di gestione archivi.

Oggi che lo standard di macchina gestionale (nel campo ovviamente dei microcomputer) è il PC IBM, e che per applicazioni produttive di un certo impegno, la sua configurazione standard è di 256/512 kbyte e hard disk, il DB III si pone come strumento che ne utilizza al massimo le caratteristiche hard.

Con il DB III è infatti normale scrivere delle procedure che gestiscono archivi che occupano un intero hard disk da 10 mega, senza nessun problema di appesantimento delle normali operazioni sugli archivi.

### Che cosa è un DBMS, che cosa è un DBMS relazionale

Un Data Base Management System è un prodotto software intermedio tra il Programma Applicativo e gli Archivi, il cui compito è quello di manipolare gli archivi stessi per mezzo di semplici comandi inseriti nella procedura. In figura 1 vediamo uno schema che esplica questa funzione in maniera grafica e quindi molto efficace.

Con un programma «tradizionale» il dialogo tra procedura e archivi avviene in maniera diretta e questo lega rigidamente i dati ad una specifica procedura. Ovviamente non è precluso l'accesso agli stessi dati da parte di altri programmi, ma questo crea come minimo dei conflitti di «competenza» tra le procedure che spesso debbono essere risolte con la duplicazione degli archivi.

Il DBMS invece è il solo a manipolare i file, e tramite il DBMS qualsiasi applicazione può avere accesso agli stessi dati, questo rende l'applicazione il più svincolata possibile dai dati e, parallelamente, i dati stessi indipen-

**Produttore:** Ashton-Tate  
**Distributore:**  
 Editrice Italiana Software S.p.A.  
 P.zza Castello 18 - 20121 Milano  
**Prezzo:** L. 1.400.000 + IVA

denti dalla procedura. Anzi, in particolare con il DB II/DB III, l'indipendenza è tale che si possono creare e gestire archivi senza scrivere una procedura.

Per quanto riguarda l'organizzazione fisica dei Database, diremo brevemente che esistono quelli Gerarchici, in cui le relazioni tra i vari segmenti sono gestite da tabelle e puntatori, con modalità efficaci, ma un po' farragino-se, e quelli Relazionali.

Nei DB relazionali ciascun archivio può essere visto come una tabella di righe (i record) e colonne (i campi), e ogni colonna può essere messa in relazione con altri archivi. In questa maniera è possibile accedere, dall'applicazione, ad uno o più archivi (relazionati tra di loro) come se fosse un unico archivio costruito proprio per le necessità dell'applicazione.

I prodotti DB II e DB III, sono a tutti gli effetti dei generatori di Data Base di tipo relazionale, per cui è im-

possibile descriverne le caratteristiche senza aver capito che cosa sia una relazione tra archivi.

### Dal DB II al DB III

Con la nascita della tecnologia 16 bit, e con la conseguente nascita dello standard IBM, il mondo delle software house si è trovato di fronte al dilemma costituito dalla scelta tra il «tradurre» i propri prodotti per le macchine 16 bit e l'investire risorse del tutto nuove sulla nuova area tecnologica.

In un primo periodo è prevalsa la prima soluzione che ha avuto il vantaggio di permettere la pressoché immediata disponibilità di numerosi pacchetti, che però non sfruttavano fino in fondo le capacità delle nuove macchine. Solo successivamente quindi sono stati messi a punto quei prodotti avanzati, di prestazioni adeguate all'ambiente hardware in cui debbono operare.

Inoltre i nuovi prodotti, essendo destinati a essere operativi su macchine «produttive» per definizione, e nascendo in un mercato ormai pieno di centinaia di pacchetti apparentemente simili gli uni agli altri, debbono raggiungere elevati livelli qualitativi che

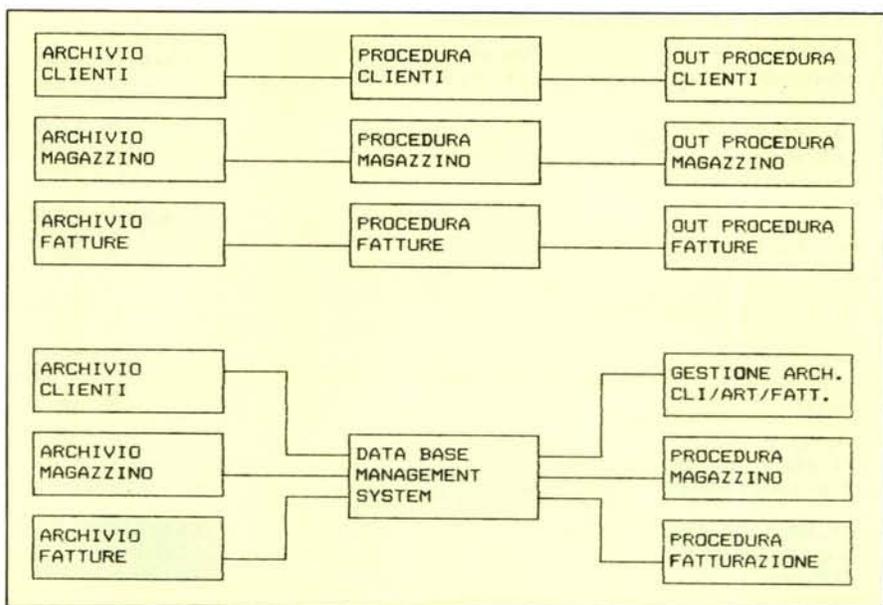


Figura 1 - Schema Descrittivo della Funzione del DBMS. Il DBMS (traducibile in Italiano in Sistema di Gestione di Base Dati) realizza la massima indipendenza tra ARCHIVI e APPLICAZIONI.

possono essere garantiti solo da software house sufficientemente grandi ed organizzate come vere e proprie case editrici.

Numerose case non hanno retto l'impatto con la nuova realtà e hanno interrotto l'attività, oppure sono state assorbite da strutture più grandi, altre, essendo nate successivamente, sono state costruite secondo le nuove regole produttive, altre, ed è il caso della Ashton Tate, sono riuscite a ristrutturarsi ed ad essere produttive anche sul nuovo fronte con prodotti totalmente ridisegnati per il nuovo ambiente hard, ma che non abbandonano le scelte strategiche vincenti che hanno determinato il loro successo nel mondo 8 bit.

È il caso appunto della Ashton Tate che ha in catalogo sia il DB II che il DB III, il cui successo sta nel fatto di essere dei Data Base Management System di elevate prestazioni, il primo

disegnato su ambiente 8 bit e trasferito sul 16 bit, il secondo progettato direttamente per questo ambiente e quindi in grado di sfruttarne appieno la possibilità, pur ricalcando la filosofia del predecessore.

Quindi la Ashton Tate ha scelto di non abbandonare una linea risultata di successo, anzi, dimostra di considerare il DB III come una nuova release del DB II, fornendo i programmi di conversione tra i due ambienti, e dimostra di considerare i file prodotti dal DB III come file standard, leggibili direttamente dal Framework, prodotto integrato della stessa Ashton Tate.

La conferma del successo di questo pacchetto viene anche dal fatto che nascono centinaia di prodotti di supporto, sia nel campo editoriale (libri, manuali, ecc.), sia nel campo software (compilatori, generatori di report, ecc.), sia infine nel campo applicativo

dove ormai sono disponibili pacchetti scritti in DB III.

## La installazione

La confezione nella quale viene fornito il DB III è una scatola di cartone telato che contiene il voluminoso manuale a fogli mobili. All'interno del manuale, c'è la guida di consultazione, comodo libretto a spirale per la rapida lettura della sintassi dei vari comandi. C'è poi un altro librettino DBASE BRIDGE che mette direttamente in luce le differenze tra i due linguaggi, e quindi risulta particolarmente utile per chi conoscendo già il DB II voglia «bruciare le tappe» nel passaggio al DB III. Essendo in realtà pieno di esempi conviene comunque leggerlo.

In un foglio mobile di plastica sono inseriti i quattro dischetti a corredo. Sono due DISCO SISTEMA 1, un DISCO SISTEMA 2 e un disco di

## Esempio di archivi in relazione tra di loro

L'esempio è composto dal programma DB III e dall'output prodotto, consistente nella visualizzazione dei tre file base, nonché da un output in cui appaiono, insieme, dati provenienti dai tre file messi in relazione tra di loro.

I comandi sono abbreviati a quattro caratteri, il che è consentito dalla sintassi del DB III.

I campi del primo file (DEBITI.DBF) sono Nome, Città, Codice, Importo. Il secondo file (TELEFO.DBF) contiene Nome, Prefisso, Telefono, ed è visto secondo un indice basato sul nome (TELEFO.NDX). Il terzo file (CODICI.DBF) contiene la scodifica dei codici presenti nel primo file, ed è quindi anch'esso indicizzato (CODICI.NDX).

In pratica il secondo archivio integra il contenuto del primo, mentre il terzo produce una tabella di decodifica dei codici.

Infine il programma, dopo essersi posizionato sul primo archivio, lo scorre tutto (con l'istruzione DO WHILE.. ENDDO) eseguendo un loop la cui condizione di fine è la fine del File. Visualizza alcuni campi del file, e tramite due relazioni, «preleva» dai due altri archivi attivi, i due campi voluti.

La sintassi del comando è SET RELATION TO X INTO Y. Dove X è il nome del campo dell'archivio chiamante, cui deve corrispondere il campo chiave dell'archivio chiamato Y. Stabili-

```
* PROGRAMMA DI PROVA
CLEA ALL
USE DEBITI
LIST
SELE 2
USE TELEFO INDEX TELEFO
LIST
SELE 3
USE CODICI INDEX CODICI
LIST
SELE 1
? "Nome      Prf. Telefono Importo Categoria"
GO TOP
DO WHIL .NOT. EOF()
? NOME
SET RELA TO NOME INTO B
?? B->PREFISSO, B->TELEFONO
?? str(IMPORTO,9)+" "
SET RELA TO CODICE INTO C
?? C->VALORE
SKIP
ENDD
?
```

ta la relazione in pratica si possono visualizzare contemporaneamente e senza operazioni intermedie campi dell'archivio chiamante, più i campi dell'archivio chiamato preceduti da un Y->, questo senza dover eseguire passaggio di variabili.

La potenza di queste funzionalità sta nel fatto che il tutto è istantaneo (cosa che si può notare solo eseguendo una prova sulla macchina), in quanto potendo lavorare su più archivi, contemporaneamente aperti, il passaggio tra l'uno e l'altro, non comporta chiusure e/o aperture di file, operazioni comunque lente. Anzi l'impressione che si ha, è che i dati siano lì pronti per essere visualizzati in quella forma.

Va notato infine che l'assenza della relazione non provoca un errore, ma semplicemente rimane un vuoto.

Record	NOME	CITTA	CODICE	IMPORTO
1	FEDERICO	ROMA	02	600000
2	LUIGI	MILANO	09	680000
3	GUIDO	GENOVA	04	345000
4	ROMEO	CATANIA	05	45000
5	MARCELLO	FIRENZE	05	300000

Record	NOME	PREFISSO	TELEFONO
1	FEDERICO	06	7312345
3	GUIDO	051	543243
2	LUIGI	02	9343211
4	ROMEO	066	456632

Record	CODICE	VALORE
1	01	MERCERIA
2	02	CALZATURE
3	03	ABBIGLIAMENTO
4	04	FERRAMENTA
5	05	FRUTTERIA
6	06	PIZZERIA
7	07	BARBIERE
8	08	CARTOLERIA
9	09	MACELLERIA
10	10	LIBRERIA

Nome	Prf.	Telefono	Importo	Categoria
FEDERICO	06	7312345	600000	CALZATURE
LUIGI	02	9343211	680000	MACELLERIA
GUIDO	051	543243	345000	FERRAMENTA
ROMEO	066	456632	45000	FRUTTERIA
MARCELLO			300000	FRUTTERIA

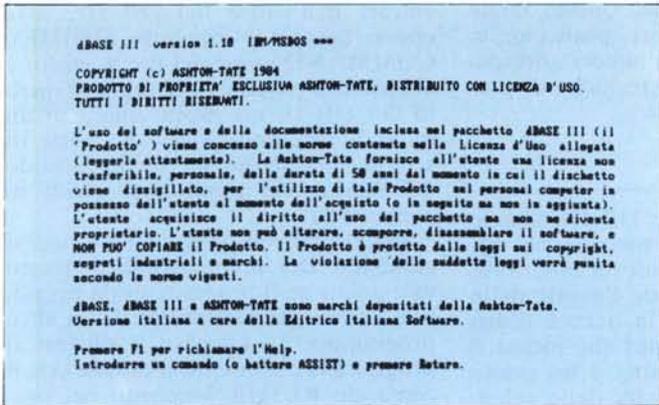


Figura 2 - Videata Iniziale del DB III. Anche il DB III, come ormai tutti i pacchetti della sua classe (e costo), non vengono venduti, ma rilasciati in licenza d'uso per 50 anni. Questo comporta da parte del licenziatario il rispetto di determinate regole di comportamento.

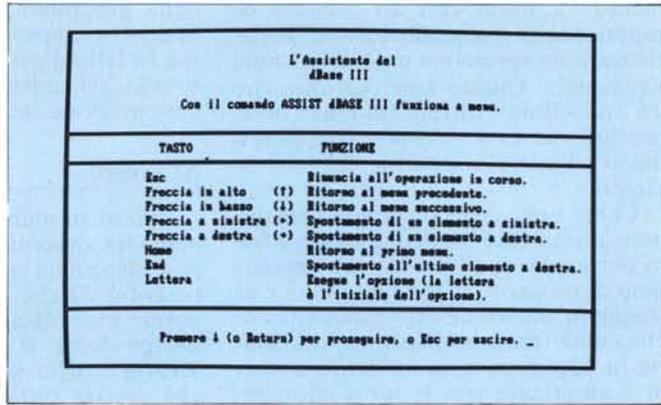


Figura 3 - Videata Iniziale dell'ASSISTANT. La prima videata dell'ASSISTANT, spiega il modo di muoversi sui menu.

ESEMPI E PROGRAMMI AUSILIARI. I primi due non sono copiabili, il terzo e quarto invece lo sono.

In una situazione di lavoro a due dischetti, il caricamento iniziale si fa, ovviamente partendo da DOS, con il DISCO SISTEMA 1, che richiede dopo qualche secondo il caricamento del DISCO SISTEMA 2 che «ci conduce» in ambiente DB. Il disco 1, quello che non si può copiare, non deve essere presente e quindi può essere nel frattempo utilizzato in altri computer.

Il sistema di protezione è quindi quello che comporta necessariamente la presenza del dischetto originale solo nella fase di caricamento iniziale.

Nel DISCO SISTEMA 2 sono presenti oltre all'overlay del DB, anche l'HELP e l'ASSIST (che come vedremo sono interattivi), non c'è molto spazio per altre cose e quindi è bene che il lavoro sia sul disco B. In una situazione di applicativo, non essendo necessario né HELP né ASSIST in linea, conviene caricare tutti i programmi sul disco A e lasciare tutto il disco B a disposizione degli archivi. Questo va bene finché l'applicativo non è molto complesso, ma se è voluminoso è probabile che lo siano anche gli archivi e che quindi occorra lavorare comunque con hard disk.

Se si lavora con hard disk si può installare tutto su una directory, seguendo la procedura di installazione, in questo modo si può accedere all'ambiente DB III direttamente, senza utilizzare nessuno dei DISCHI SISTEMA. L'installazione su hard disk comporta il fatto che il DISCO SISTEMA 1 non potrà più essere utilizzato se non dopo aver eseguito la procedura inversa di disinstallazione.

A completamento dell'installazione occorre inserire obbligatoriamente nell'unità con la quale viene lanciato il DOS il file CONFIG.SYS, con il quale il DOS setta il numero dei file e dei buffer di lavoro, e che il DB III sfrutta

al massimo. Nel disco dove c'è il DB overlay può essere presente il file CONFIG.DB, che si costruisce (così come il CONFIG.SYS) sotto DOS, che setta il modo di lavorare del DB.

Ad esempio con il CONFIG.DB si può settare l'uso del colore, e questo comporta la definizione dei colori sfondo, primo piano, zone di input. La definizione vale fino a che non venga eseguito un nuovo settaggio. Con il CONFIG si può anche indicare il PROGRAMMA applicativo da caricare direttamente con il boot.

### Il manuale

Il manuale è suddiviso in varie parti per un totale di circa 500 pagine. La prima parte si chiama Apprendimento e descrive le varie funzioni in maniera graduale e discorsiva. La seconda parte si chiama Comandi (è quindi il Reference) ovvero l'elenco dei comandi, in ordine alfabetico, uno per pagina con la sintassi e un esempio.

Vi è poi la parte relativa alle funzioni, che sono di sistema, matematiche, di stringa e di data, ed infine una serie di appendici e di tabelle esplicative.

È senza dubbio fatto molto bene, non solo come organizzazione e come testo, ma anche da un punto di vista «editoriale», che peraltro è identico a quello dell'originale americano. La traduzione realizzata a cura della Editrice Italiana Software è molto accurata e sicuramente è stata fatta da specialisti.

L'approccio via manuale al DB III va fatto leggendo la parte «apprendimento». Una volta appreso l'argomento durante l'uso dovrebbe essere sufficiente ricorrere all'HELP interattivo per ricordare la sintassi delle funzioni (di cui però bisogna ricordare l'esistenza) oppure, se per motivi di spazio non si è caricato l'HELP, ricorrendo al prontuario. In casi estremi si deve ricorrere al REFERENCE dove il sin-

golo comando è spiegato al di là della sintassi.

Una cosa che manca, e che invece era presente sul manuale «americano», sono i listati dei vari programmi dimostrativi. Il quarto dischetto contiene infatti un vero e proprio programma di contabilità che al di là della sua funzionalità, che può non interessare, serve da esempio «reale» di utilizzazione del linguaggio e come tale dovrebbe essere commentato.

### I modi di lavorare del DB III

Il DB III può essere utilizzato in due modalità differenti, in comandi diretti, oppure come vero e proprio linguaggio di programmazione con il quale costruire delle procedure.

Con la prima modalità si possono costruire e gestire archivi, crearne varie organizzazioni, disegnare dei Report, operare selezioni basate su confronti logici, ecc. Si riesce anche a lavorare su più archivi utilizzando relazioni.

Si arriva quindi a operare su archivi consistenti, senza dover scrivere procedure o addirittura senza saper programmare. Se questo tipo di utilizzazione è sufficiente si può anche sfruttare l'ASSIST, che è un'interfaccia tra utilizzatore e prodotto, organizzato secondo la ormai consolidata tecnica dell'albero dei menu e che permette di operare su molte funzionalità (che vedremo dopo) guardando solo il video e dirigendo le operazioni tramite due tasti freccia, e contemporaneamente... imparando il DB III.

Ma il DB III è, oltre a un Data Base Management System, anche un potente linguaggio di programmazione, con il quale è possibile realizzare procedure anche molto complesse e che possono sfruttare al massimo le elevate potenzialità di un personal computer e del suo hard disk.

È chiaro che in questo caso lo stru-

mento va usato con un minimo di esperienza di programmazione. Esperienza o da specialisti o da utilizzatori «avanzati». Questo non vuol dire che un «novellino» troverà un muro invalicabile, ma vuol solo dire che, perlomeno all'inizio, sottoutilizzerà lo strumento.

Come linguaggio di programmazione è ovviamente particolarmente adatto per quelle applicazioni che necessitano di numerosi archivi ed indici e di frequenti accessi ad essi. È potente anche come funzioni di stringa, necessarie in applicazioni gestionali, ma non lo è altrettanto con le funzioni matematiche, per cui risulta inadatto per applicazioni scientifiche (mancano ad esempio del tutto le funzioni trigonometriche), ed inoltre, essendo un interprete risulta molto lento nei calcoli matematici complessi.

Il DB III ha una sintassi che «obbliga» a fare programmazione strutturata, soprattutto per il fatto che mancano del tutto le istruzioni di salto, che sono state sempre avversate dai puristi

della programmazione. Questo mette in crisi i programmatori «pasticcioni», ma fa felici i programmatori «ordinati» che già utilizzano tecniche di programmazione strutturata.

### Al lavoro

Entrati in ambiente DB III, con le modalità descritte in precedenza, appare dapprima una videata monitoria (vedi fig. 2) che avverte l'utente delle norme che regolano la licenza d'uso del prodotto. Il prompt che indica il READY della macchina è un punto che appare sulla sinistra dello schermo.

A questo punto si può usare il DB III in comandi diretti, che permettono quasi tutte le funzionalità sugli archivi (ovvero, creazione, aggiornamento, cancellazione, ecc.), permettono l'uso dei comandi di servizio (indice dei file, copia, cancellazione e rinominazione dei file, ecc.), permettono la definizione delle variabili, ecc.

Per scrivere un programma occorre

entrare nell'editor del DB III, cosa che si fa con il comando MODIFY COMMAND nome del file. L'editor è abbastanza potente (molto più di quello del DB II, ma molto meno di un buon Word Processor) e permette di scrivere il programma anche facendo dei merge con programmi scritti in precedenza.

Per eseguire il programma si usa il comando DO nome del file. Questo vale anche se il programma da eseguire è a sua volta richiamato da un altro programma. In tal caso il ritorno al programma chiamante si ottiene con il comando RETURN inserito nel programma al livello più basso.

Interessante è la gestione delle variabili che sono rilasciate automaticamente al rientro nel programma chiamante, a meno che non siano definite, con un'apposita istruzione, pubbliche.

Questa regola, unita al fatto che il numero delle variabili definibili è di 256 e che la loro memoria complessiva può essere ampliata in fase di settaggio dei parametri, pone rimedio a quello che era uno dei maggiori difetti del DB II, cioè la limitazione sul numero e la memoria totale disponibile per le variabili.

Anche se si deve lavorare con programmi, la creazione degli archivi va fatta tramite il comando diretto CREATE che utilizza una maschera per l'immissione dei campi, del tipo dei campi (vedi dopo) e della loro lunghezza.

Una volta creato il file, ma sempre lavorando in comandi diretti, si possono eseguire i comandi APPEND (per l'accodamento dei record), EDIT per la loro eventuale modifica, BROWSE che mostra gli stessi dati su più righe e quindi permette modifiche sull'archivio con modalità FULL SCREEN. Inoltre si possono eseguire tutti i comandi diretti di ricerca, selezione, cancellazione, nonché modifica per condizione.

Circa il modo con il quale ci si può muovere sull'archivio, rimandiamo al paragrafo seguente intitolato File Dati e File Indici.

### Il comando Assistant

Il metodo più veloce per apprendere il DB III, per chi non conosca il DB II, è quello di usare l'ASSISTANT. L'ASSISTANT è un tool che permette di utilizzare buona parte dei comandi del DB III per mezzo di un menu guidato. E mentre si eseguono i comandi, e quindi se ne comprende il funzionamento, appaiono anche delle righe di testo che spiegano il comando stesso.

Nelle figure 3 e 4 vediamo i primi due menu, quello introduttivo, che

FUNZIONALITA' DELL' ASSISTANT DB III		
COMANDI ASSISTANT		COMANDI DB III
1.1iv.	2.1iv.	
IMPOSTA	Apri	:USE file dati INDEX file indici
	Drive	:SET DEFAULT TO drive
	Crea	:CREATE nome file
	Crea Etichetta	:CREATE LABEL nome label
	Crea Prospetto	:CREATE REPORT nome report
MODIFICA	Aggiunge	:APPEND
	Corregge	:BROWSE
	Edit	:EDIT numero record
	Cancella	:DELETE numero record FOR condizioni
	Ripristina	:RECALL numero record FOR condizioni
	Sostituisce	:REPLACE campi WITH espressioni
	Posiziona	: vedi POSIZIONA
POSIZIONA	Cerca	:LOCATE FOR condizioni
	Trova	:FIND chiave
	Prosegue	:CONTINUE
	Salta	:SKIP numero passi
	Va	:GO numero record
	Modifica	: vedi MODIFICA
	Recupera	: vedi RECUPERA
RECUPERA	Visualizza	:DISPLAY campi FOR condizioni
	Somma	:SUM campo FOR condizioni
	Media	:AVERAGE campo FOR condizioni
	Conta	:COUNT campo FOR condizioni
	Etichette	:LABEL FORM nome label FOR condizioni
	Prospetto	:REPORT FORM nome report FOR condizioni
	Posiziona	: vedi POSIZIONA
ORGANIZZA	Crea Indice	:INDEX ON nome campo TO nome file indice
	Ordina	:SORT ON nome campo TO nome file
	Copia	:COPY campi TO nome file FOR condizioni
	Compatta	:PACK
SERVIZI	Drive	:SET DEFAULT TO drive
	Copia File	:COPY FILE nome file 1 TO nome file 2
	Indice	:DIR
	Rinomina	:RENAME nome file 1 TO nome file 2
	Elimina	:DELETE FILE nome file
	Modif. Struttura	:MODIFY STRUCTURE

Figura 5 - Schema Comparativo tra ASSISTANT e COMANDI DB III. L'ASSISTANT costituisce una interfaccia tra molti dei comandi DB III e l'utente alle prime armi che vi accede logicamente tramite un menu guidato.

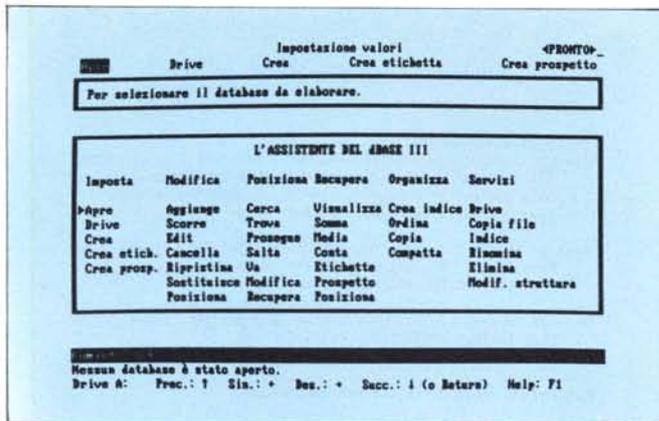


Figura 4 - Help Iniziale sulla Maschera dell'ASSISTANT. Anche l'ASSISTANT permette l'accesso all'HELP interattivo... più facile di così... Nel nostro caso si vede l'indice dei menu di secondo livello.

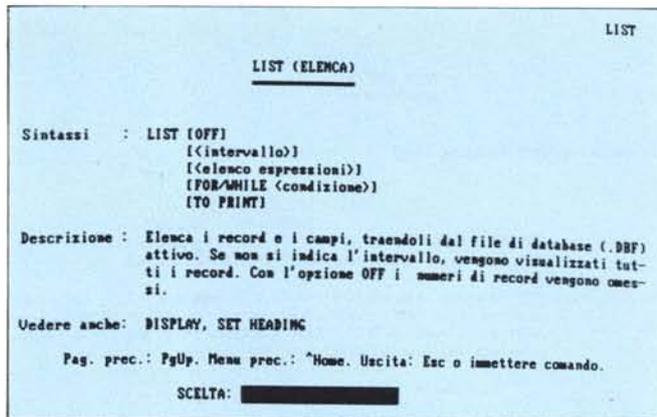


Figura 6 - Videata di Help Comando LIST. L'HELP fornisce la sintassi del comando richiamato. È molto utile in quanto alcuni comandi hanno sintassi molto articolate in quanto permettono numerose e differenti funzionalità.

spiega i vari movimenti possibili, e quello principale, che indirizza alle prime grandi categorie di comandi.

In figura 5 riportiamo una tabella che mette in relazione le funzionalità dei due livelli di menu dell'Assistant con i corrispondenti comandi DB III, richiamati per eseguire la funzionalità stessa, e, poiché i comandi di menu sono in Italiano e quelli DB III rimangono in Inglese, si fa contemporaneamente un buon esercizio di traduzione.

È quindi un metodo di apprendimento interattivo e guidato, ma non può arrivare a sfruttare fino in fondo, né lo vuole, le potenzialità del DB III. Inoltre essendo in pratica esterno al DB III la sua presenza sul dischetto operativo non è indispensabile.

### Tipologia dei comandi e loro sintassi

Il DB III possiede circa 150 comandi, divisibili grossolanamente in alcuni gruppi logici:

- creazione di file dati (definizione di una struttura)
- gestione di file (senza modificare la struttura)
- manipolazione di file dati (uno o più file)
- editazione dei file (per visualizzazione e/o modifica)
- elaborazione
- comandi di servizio (accesso anche a comandi DOS)
- gestione delle variabili di memoria
- parametri da settare (35 parametri)
- gestione di altri tipi di file (REPORT, LABEL)
- comandi utilizzabili in programmi
- comandi per gestire le periferiche.

Dispone inoltre di circa 50 funzioni, tra matematiche, logiche, di stringa e di data.

Alcuni comandi sono estremamente potenti e necessariamente la loro sintassi diventa complessa. In pratica con un unico comando si riesce a produrre un effetto che con un linguaggio «tradizionale» avrebbe richiesto numerose righe di programma. Facciamo due esempi.

Il comando LIST. Il comando più semplice per visualizzare un archivio (già creato e contenente dati) è il LIST, che ha una sintassi molto flessibile in funzione del risultato che si vuole ottenere (vedi fig. 6). Esempio:

LIST elenca tutti i campi di tutti i record.

LIST OFF come sopra, senza la visualizzazione del numero record (gestito da DB III)

LIST <elenco dei campi> visualizza solo i campi desiderati, inoltre all'interno dell'elenco dei campi si possono inserire anche altre funzioni (ad esempio il prodotto di 2 campi).

LIST FOR <condizione>. Dove la condizione è una qualsiasi composizione di condizioni logiche eseguite su uno o più campi.

Inoltre è possibile, tramite un settaggio iniziale, definire l'ordine di visualizzazione (SET INDEX TO...), oppure decidere se far apparire o meno le intestazioni dei campi (SET HEADING ON/OFF).

La sintassi del comando ha quindi una parte obbligatoria, nel nostro caso LIST, e una serie di parti facoltative che permettono di «tagliare su misura» il risultato.

Vediamo il comando COPY (fig. 7). Il comando COPY ha una sintassi simile al LIST solo che il risultato non è un output su video o stampante, ma su file. O meglio viene creato, direttamente dal comando COPY, un file di

struttura pari a quella dei campi passati e contenente solo i record che rispettano le condizioni impostate.

Con un linguaggio tradizionale lo stesso risultato si può ottenere solo creando prima il file «ricevente» e scrivendo una routine per eseguire il «filtro» dei dati.

I comandi più complessi sono quelli di tipo CREATE, che permettendo di realizzare o una struttura o un prospetto, comportano l'esecuzione di vere e proprie sottoprocedure.

In figura 8 vediamo la maschera prodotta dal comando CREATE. L'immissione dei vari campi comporta la definizione del nome, del tipo (per la tipologia dei dati vedi dopo), della lunghezza. Il comando esegue i vari controlli formali sui dati immessi ed esegue i conteggi.

Un interessante miglioramento rispetto al DB II è l'esecuzione del MODIFY STRUCTURE, direttamente sull'archivio da modificare. Questa operazione avviene conservando tutti i dati preesistenti, i cui campi non vengono modificati.

Un altro comando estremamente potente, al di là di quanto faccia capire lo stesso manuale, è il CREATE REPORT, di cui in figura 9 vediamo la maschera operativa, che permette di controllare sia i campi dell'archivio del quale si sta realizzando il REPORT, sia direttamente il formato risultante.

Il comando permette non solo la realizzazione di totali e sottototali (ovviamente per far questo l'archivio deve essere indicizzato secondo i campi di raggruppamento), ma di realizzare al suo interno una relazione con un altro archivio e con variabili esterne.

Il CREATE REPORT crea solo un formato di stampa, ovvero un modo di stampare certi dati. Per richiamare tale formato c'è il comando REPORT FORM nome del formato <condizio-

```

COPY (COPIA)

Sintassi :COPY TO <nome file> [(intervallo)]
          [FIELDS <elenco campi>]
          [FOR/WHILE <condizione>]
          [SDF] / [DELIMITED [WITH
          BLANK/<delimitazioni>]]
COPY STRUCTURE TO <nome file> [FIELDS <elenco campi>]
COPY TO <nome file di struttura> STRUCTURE EXTENDED

Descrizione : COPY TO genera una copia di un file di database (.DBF) attivo su
un altro file .DBF, oppure (con le opzioni SDF e DELIMITED) su
un file di testo (.TXT). COPY STRUCTURE crea una copia della
struttura del file di database (.DBF) attivo.

          Premere Return per continuare.

          SCELTA: ██████████

```

Figura 7 - Videata di Help Comando COPY. Il comando COPY è uno dei più potenti, in quanto permette «in un colpo solo» di creare partendo da un archivio attivo, un altro archivio, stabilendo sia i campi sia i record da inserirvi.

```

. disp stat

Database selezionato:
Area di lavoro: 1, database attivo: B:opdat.dbf, alias: OPDMT
File indice: B:opnd1.ndx, chiave: ocl1+otab+dtoc(odat)
File indice: B:opnd2.ndx, chiave: otab+ocli+dtoc(odat)
File indice: B:opnd3.ndx, chiave: oprg

Area di lavoro: 2, database attivo: B:clndt.dbf, alias: CLDMT
File indice: B:clnd1.ndx, chiave: cdox
File indice: B:clnd2.ndx, chiave: cnox

Area di lavoro: 3, database attivo: B:ttndt.dbf, alias: TTDMT
File indice: B:ttnd1.ndx, chiave: tabi
File indice: B:ttnd2.ndx, chiave: tnox

          Premere un tasto qualsiasi per continuare.

```

Figura 8 - Maschera di Lavoro del Comando CREATE. Il comando CREATE, nome del file, è quello con cui si crea la struttura di un Archivio, le operazioni sono guidate da una maschera che ne controlla la correttezza formale.

ni> <output>. Ovvero definito il come stampare (con il formato), con le condizioni si stabilisce che cosa stampare e su quale output (PRINTER, SCREEN, FILE).

E poiché inoltre le condizioni possono essere fornite via variabile, con un'unica istruzione si possono eseguire numerose stampe, ciascuna delle quali molto complessa.

### Tipi di file

Lavorare con il DB III comporta, volenti o nolenti, la scrittura di numerosi file, che il DB III stesso battezza con ben 9 tipi differenti di suffissi.

.DBF	File Dati
.DBT	File con Campi MEMO
.NDX	File Indici
.FMT	File Formato Video
.FRM	File Formato Stampa
.LBL	File Formato Etichette
.MEM	File Variabili di Memoria
.PRG	File di Comandi o Procedura
.TXT	File di Testi in Formato ASCII

La costruzione di un archivio comporta la scrittura di un file tipo \*.DBF, se uno dei campi di questo file è del tipo MEMO, sarà creato anche un file tipo \*.DBT.

Se il file deve essere organizzato secondo una o più chiavi, si realizzeranno uno o più file tipo \*.NDX.

Se l'accesso ai file deve avvenire attraverso una maschera di immissione, disegnata dall'utente (non quella standard proposta da DB III) si costruisce un file tipo \*.FMT.

In output si possono costruire, e il DB III guida la costruzione in maniera interattiva, REPORT, basati non solo sui campi di un archivio, ma anche su relazioni con altri archivi e a calcoli matematici con variabili esterne all'archivio. Il file generato è di tipo

\*.FRM, oppure \*.LBL se si stampano etichette.

La procedura, scritta in linguaggio DB III ha suffisso \*.PRG. Tale formato è leggibile da DOS e anche da Word Star. In pratica per scrivere una procedura in DB III, non è indispensabile entrare in DB III.

Infine un file \*.DBF può essere «tradotto» in formato ASCII standard e come tale letto direttamente da altri linguaggi o prodotti che hanno accesso al disco su file di questo tipo (es. BASIC, SYMPHONY, WORDSTAR).

Va anche detto che il DB III ha «ottimi rapporti» con il DOS. Possiede infatti un comando RUN richiamabile con il punto esclamativo e con il quale si può eseguire un comando DOS (che abbia desinenza \*.COM oppure \*.EXE). Questo a condizione che con il settaggio tramite il CONFIG.DB si lasci spazio per altre applicazioni.

È quindi possibile dal DB III, creare un file \*.TXT leggibile da Basic, sempre dal DB III, richiamare il Basic ed eseguire un programma Basic che rielabora il file \*.TXT, e alla fine ritornare al DB III. Niente male!

### Tipi di campi

Durante la creazione di un archivio è possibile definire 5 tipi di campi.

- C Carattere (fino a 254 caratteri).
- N Numerico (fino a 15 cifre significative, e con possibilità di definire la posizione della virgola).
- D Data nel formato europeo (GG/MM/AA). Tale formato consente l'esecuzione di calcoli numerici e di confronti logici. Per la visualizzazione occorre usare delle funzioni di traduzioni in stringa.

M Memo (fino a 4.000 caratteri in forma libera).

L Dati di Tipo Logico (S s T t e il contrario N n F f).

Ottimo il formato DATA, che fortunatamente è stato tradotto in formato europeo, anche se esiste la possibilità di usare (ma solo in fase di output) molti altri formati. Buona l'idea dei campi formato MEMO, ma è troppo difficile usarli produttivamente, in quanto non è possibile eseguire su di essi operazioni di editing né in fase di immissione, né in fase di emissione.

Non ho particolare propensione per i campi logici, pressoché equivalenti a campi FLAG di un solo carattere. Inoltre, e questo lo considero un difetto, l'operazione di confronto logico, produce una variabile logica T-true (vero) o F-false (non vero) e non un valore matematico, più direttamente usabile in calcoli matematici. Questo conferma la scarsa attitudine del DB III ai calcoli aritmetici.

### File dati e file indici

Una delle caratteristiche principali del DB III è costituita dalla suddivisione degli archivi in archivi DATI, che sono organizzati in modo sequenziale, e in archivi INDICI, che costituiscono, potremmo dire, un modo di vedere l'archivio DATI.

Il DB III attribuisce ad ogni record un numero progressivo di immissione che si può richiamare con la funzione RECNO(). L'assegnazione di questo numero è quindi sequenziale, ed è gestita dal DB III. Per ricercare un record in un file bisogna o conoscerne il numero progressivo (e si usa la istruzione GO N), oppure occorre ricercarlo scorrendo tutto il file fin quando non si trova quello che corrisponde ad

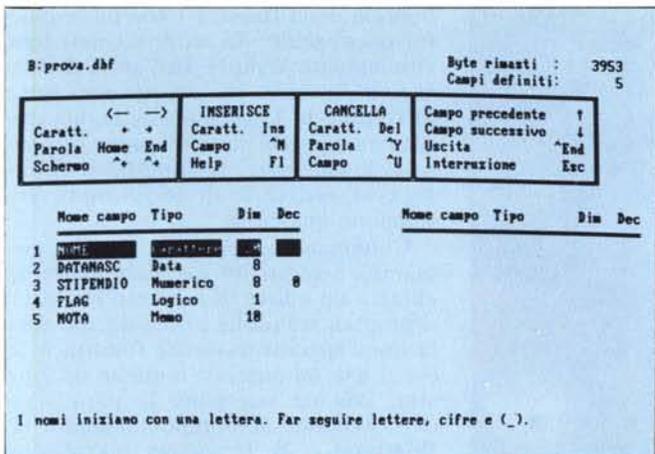


Figura 9 - Maschera di Lavoro del Comando CREATE REPORT. Il comando CREATE REPORT costituisce un vero e proprio Report Generator, con il quale in maniera controllata, si può costruire un Report anche complesso che visualizza ed elabora dati provenienti da più files.

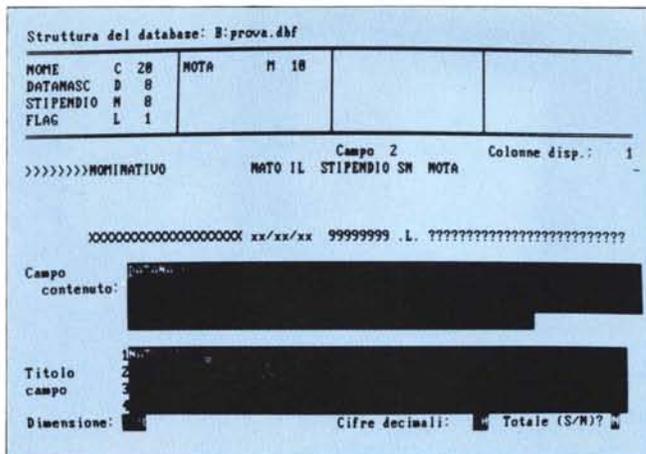


Figura 10 - Videata conseguente al Comando DISPLAY STATUS. Quando si lavora contemporaneamente su più file, alcuni dei quali indicizzati, è necessario sapere esattamente quale sia quello selezionato al momento.

una data condizione. Se l'archivio è lungo occorrono minuti.

Le uniche operazioni che riattribuiscono i numeri progressivi sono il SORT (ordinamento fisico del file), il PACK (cancellazione fisica dei record cancellati logicamente), e l'INSERT (immissione di un record con un numero determinato). Sono quindi tutte operazioni lente, assolutamente da evitare all'interno di procedure gestionali, specialmente se gli archivi sono grandi.

Il FILE indice dà la soluzione a questi problemi in quanto consiste in un archivio, in genere ben più ristretto di quello DATI, in cui una determinata chiave di ricerca (ad esempio un campo dell'archivio DATI) viene legata al progressivo gestito dal DB III. E la ricerca tramite questa chiave è comunque istantanea, qualsiasi sia la dimensione dell'archivio dati.

In pratica dato un Archivio Dati si costruisce una o più modalità di lettura (ARCHIVI INDICI) per cui è possibile tramite una o l'altra chiave accedere direttamente al record cercato, tramite la istruzione FIND «chiave».

Una volta definita una o più chiavi di accesso, il DB III garantisce comunque l'allineamento dei vari archivi (quello DATI e quelli INDICI) durante le varie fasi operative sul file.

Il concetto di FILE DATI e FILE INDICE è fondamentale, anche se purtroppo non è un concetto facile.

### Valutazioni e conclusioni

Il DB III è un Data Base Management System e quindi risolve al meglio problemi di gestione di grossi volumi di dati, è quindi adatto ad essere utilizzato in quelle applicazioni che manipolano grossi volumi di dati. È un Data Base Relazionale e quindi permette un accesso contemporaneo a più

archivi, che ai fini dell'applicazione possono essere considerati come un'unica base dati.

È un prodotto destinato ad essere utilizzato per applicazioni importanti e quindi da gente esperta. È comunque prevista anche un'utilizzazione «friendly», ma questa avviene tramite un'interfaccia esterna, l'Assistant, che in una procedura operativa e produttiva non serve.

Per utilizzazioni importanti diventa un linguaggio di programmazione a tutti gli effetti, con il quale è possibile realizzare procedure destinate a utenti

### Faremo un corso pratico di DB II e DB III

MCmicrocomputer pubblicherà un corso pratico di DB II e DB III, che inizierà dal prossimo numero e che sarà curato da Francesco Petroni. Lo definiremo pratico in quanto non vuole essere un doppiopione del manuale, magari messo in altra forma, manuale che come al solito rimane il principale testo di ogni prodotto. Ma vuole essere un approfondimento basato soprattutto su esempi pratici che facciano capire meglio le varie funzionalità e che eventualmente ne suggeriscano ambiti applicativi.

Un'altra condizione sulla quale fonderemo il corso è quella di non richiedere alcuna pre conoscenza di altri linguaggi in quanto le varie funzionalità saranno spiegate con esempi, in tal modo cercheremo anche di «raggiungere» lettori che attualmente non usano DB II o DB III, ma che avendone sentito spesso parlare vogliono vedere di che cosa si tratta.

Il corso sarà contemporaneamente di DB II e DB III, data la identica filosofia e la quasi identica sintassi dei due linguaggi. Ove fosse necessario citeremo e faremo vedere le differenze.

finali non specialisti. La stesura dei programmi è facilitata sia da una logica di programmazione che costringe alla programmazione strutturata, sia dall'esistenza di funzioni così potenti che da sole risolvono buona parte dei problemi abituali.

Il DB III inoltre può produrre file del tipo TXT, leggibili direttamente da DOS e quindi dagli altri linguaggi, e questo risolve buona parte dei problemi di integrazione con procedure già esistenti e/o future.

Quindi dei due obiettivi che aveva l'Ashton Tate, quello di sfruttare al massimo le potenzialità di un microcomputer 16 bit della classe PC IBM, è sicuramente raggiunto. Quello di rinverdire il successo commerciale del DB II, con un prodotto che ne ricalca la filosofia (e ne corregge i difetti) si potrà giudicare fra qualche mese.

Mentre del DB II non esiste una versione italiana, il DB III esce, abbastanza tempestivamente rispetto alla analoga release americana, in italiano e in una veste editoriale identica all'originale. La versione è curata dalla Editrice Italiana Software, che rappresenta in Italia i prodotti Ashton Tate e che quindi ha in catalogo anche il Framework, la cui prova è stata da noi pubblicata nel numero 41 di MC. Il DB III inoltre è presente nei cataloghi ufficiali sia IBM che Olivetti.

Il prezzo di 1.400.000, malgrado la potenza del prodotto e malgrado la elegante veste editoriale, è elevato, anzi addirittura proibitivo se l'utente ne deve fare una utilizzazione «personal» o hobbistica o semplicemente di studio del linguaggio.

Ma è chiaro che se il DB III serve per realizzare una procedura operativa in un reale ambiente di lavoro, la spesa diventa minima rispetto agli investimenti in termini di macchine, uomini e programmi.