



di Francesco Petroni

## Funzioni avanzate

### Sesta parte

Nel corso di questa ultima puntata del corso DB III studieremo alcune problematiche avanzate e vedremo come risolverle in ambiente DB, utilizzando sia funzioni o insiemi di funzioni già viste, sia analizzando alcune funzioni più potenti, ancora non esaminate.

Va chiarito che anche in ambiente DB dato un problema le soluzioni praticabili sono sempre e comunque tante. La soluzione scelta dipenderà dalla sensibilità dell'utilizzatore e dalla presenza o meno di funzioni specifiche che aiutino a risolverla.

Vedremo quindi una serie di problemi e vedremo come affrontarli da un punto di vista generale e poi in particolare ne vedremo soluzioni DB.

### Il Sort

Il Sort, ovvero la riorganizzazione secondo una certa chiave di ordinamento di un insieme di dati, ha appassionato centinaia di studiosi e teorici della programmazione che hanno versato fiumi di inchiostro sull'argomento.

Il problema è appassionante perché può essere del tutto astratto dalla pratica e quindi rientra nel campo della ricerca pura. Si ricerca un algoritmo per mezzo del quale da un insieme di dati comunque ordinati si ottiene un altro insieme ordinato secondo una legge, eseguendo una serie di confronti e una serie di scambi di posizioni tra coppie di elementi.

L'algoritmo migliore è quello che, a

parità di risultato, esegue meno confronti e meno scambi e quindi impiega meno tempo.

Il problema del Sort non esiste più in quanto non c'è strumento software, linguaggio tool, package che sia che non comprenda l'utility Sort. Ad esempio nei tabelloni elettronici tipo Lotus 123, il Sort si esegue indicando solo due cose, insieme dei dati e colonna contenente i dati in base alla quale ordinare.

In ambiente DB l'utilità del Sort è minima in quanto in tutti gli ambiti applicativi normali può essere sostituito in tutto e per tutto dall'indice.

Per chiarire le idee, il Sort comporta una riorganizzazione fisica dei dati secondo una vista logica, mentre l'indice lascia l'organizzazione fisica inalterata creando una vista logica attraverso la quale vedere l'archivio.

Il confronto si risolve a tutto vantaggio dell'indice perché può essere tenuto costantemente aggiornato (e il Sort no), perché essendo indipendente dalla organizzazione fisica ne possono esistere diversi (mentre l'archivio «sortato» può essere uno solo) e perché mentre è possibile creare un indice manipolando con funzioni la chiave, il Sort può essere eseguito usando uno o più campi come chiave.

Da un punto di vista più pratico il comando DB di Sort è equivalente ad un comando di copia in quanto non viene riordinato il file originale, ma ne viene eseguita una copia ordinata se-

RecId	NO	CO	SEX	DATANASC	PROVNASC	STIPENDIO
1	LUCIA	FIORENTINI	F	13/11/61	RM	1090000
2	BENIAMINO	SIALI	M	21/05/48	LI	1075000
3	NATALE	ARANCIO	M	04/09/58	LI	1450000
4	LIDIO	SIAMONDI	M	29/05/51	LI	1190000
5	GIUSEPPE	ABATINI	M	04/04/55	RM	6450000
6	ROD	LUSSU	M	22/03/65	TO	540000
7	LUIGI	BIANCHI	M	19/11/54	LI	2190000
8	CHIARA	ROSA	F	25/03/60	MI	1450000
9	DORINA	MALTESE	F	11/11/65	MI	4560000
10	CARLOTTA	LUDDVISI	F	22/11/63	MI	1875000
11	ALDO	AZZURRI	M	15/02/56	MI	9000000
12	WALTER	CHIALENTI	M	25/03/61	RM	3450000
13	ALESSANDRO	MALINI	M	12/12/56	MI	3020000
14	DONATELLO	DOZZI	M	21/01/54	TO	6430000
15	LUISA	ROSSINI	F	19/06/55	LI	5200000
16	MARGHERITA	VIOGA	F	03/10/54	RM	1000000
17	LUCIL	RUSSI	M	09/06/58	RM	1200000
18	MALTA	NERI	M	11/01/55	RM	1100000
19	BATRIGL	MARRONI	F	29/06/57	TO	1290000
20	MALTYR	LUCIOLI	M	11/02/55	RM	4100000
21	BIGI	MALETTI	M	19/02/57	RM	1295000
22	MARCO	VERDI	M	20/07/61	LI	1250000
23	MARIO	BRUNO	M	20/12/62	MI	4345000
24	CARLO	BINI	M	12/12/61	MI	1745000

Figura 1 - Elenco dei record del file di prova. Come al solito forniamo subito il listato, elenco dei record ottenuto con il comando DB List, del file che utilizzeremo per le prove. Questo per rendere più comprensibile «l'effetto» dei vari comandi che vedremo.



condo uno o più campi. La sua sintassi è:

```

SORT TO <file destinazione>
  ON <nome campo 1, nome campo 2>
  <intervallo>
  <condizione>
    
```

ON indica il primo e gli altri campi di ordinamento, mentre con l'intervallo e la condizione è possibile limitare i record copiati a quelli che rispettano certe condizioni.

La sintassi del comando di copia è:

```

COPY TO <file destinazione>
  FIELDS <nome dei campi>
  <intervallo>
  <condizioni>
  <specifiche>
    
```

Ovvero dall'archivio attivo è possibile eseguire una copia selezionando record (con le condizioni) e campi (con la specifica FIELDS). Immaginando l'archivio come una tabella righe e colonne se ne eseguono «affettamenti» verticali e orizzontali.

Se il file di cui si esegue la copia è ordinato secondo un certo indice, il risultato dell'operazione è del tutto equivalente ad un Sort sulla stessa chiave dell'indice.

Per meglio capire il modo di lavorare di tali comandi va considerato che ambedue pretendono un file attivo, ne creano uno nuovo, lavorano rimanendo sul file attivo, mentre quello creato rimane chiuso.

Va inoltre capita la differenza tra il comando COPY e il COPY FILE che, più brutalmente esegue un duplicato di un file di qualsiasi tipo, indipendentemente anche dal fatto che sia o meno aperto un file, e che il duplicato sia un file dati o un indice o un programma o altro.

Le indicazioni che invia il DB sono in numero record copiati con il comando COPY e in numero di byte copiati con il COPY FILE.

### Totalizzazioni di dati

Altro problema classico nella elaborazione dei dati è quello costituito dalla totalizzazione dei dati numerici. E anche in questo caso le strade percorribili sono numerose, volendole elencare sono almeno 4.

Utilizzo delle funzioni SUM, COUNT, AVERAGE.

Utilizzo del REPORT con opzione TOTAL.

Costruzione di un programma con loop DO.. WHILE.

Utilizzo della Funzionalità TOTAL.

1) Nel primo caso si ricorre a funzioni specifiche, molto comode se usate al volo, ma pressoché inutilizzabili in programmi. SUM esegue la somma di campi numerici, COUNT il conteggio e AVERAGE la media (vedi fig. 2). La loro sintassi è analoga:

SUM, COUNT, AVERAGE <elenco espressione numerica campi>

<condizione>  
<TO variabili>

```

- sum
  24 record sommati.
  STIPENDIO
  59004200
- sum for sex="M" to s1
  17 record sommati.
  STIPENDIO
  42430200
- sum for sex="F" to s2
  7 record sommati.
  STIPENDIO
  16566000
- aver(date()-datanasc)/365 to medieta
  24 record con media calcolata.
  (date()-datanasc)/365
  22.79
    
```

Figura 2 - Utilizzo dei comandi diretti di Somma, Media, ecc. Sono funzioni molto potenti, che comportano uno «scorrimento» dell'archivio. Durante questo scorrimento vengono eseguiti conteggi, somme, medie. Si tratta di funzioni alquanto dirette, ovvero facili e utili in comandi diretti, un po' difficili da usare in programmazione, dove si ricorre ad altre funzionalità.

L'elenco delle espressioni è un concetto più estensivo di campo. Ovvero è possibile non solo eseguire operazioni matematiche su campi, ma anche su espressioni numeriche che li comprendono. Con le condizioni è possibile limitare i record sui quali eseguire i calcoli.

TO (variabili) serve per riportare il risultato ottenuto su variabili da utilizzare in seguito.

Il funzionamento della istruzione è intuitivo, viene scorso l'archivio e viene incrementato un contatore. Il risultato non è quindi istantaneo, specie se

Pagina n.	1
01/01/80	
TOTALIZZAZIONE PROVINCIA & SESSO	
** RAGGRUPPAMENTO PROVINCIA LI	
* SESSO F	
Subtotale parziale	5200000
* SESSO M	
Subtotale parziale	7955000
Totale parziale	13155000
** RAGGRUPPAMENTO PROVINCIA MI	
* SESSO F	
Subtotale parziale	7891000
* SESSO M	
Subtotale parziale	9708000
Totale parziale	17599000
** RAGGRUPPAMENTO PROVINCIA RM	
* SESSO F	
Subtotale parziale	2180000
* SESSO M	
Subtotale parziale	17801000
Totale parziale	19981000
** RAGGRUPPAMENTO PROVINCIA TO	
* SESSO F	
Subtotale parziale	1295000
* SESSO M	
Subtotale parziale	6974200
Totale parziale	8269200
Totale	59004200

Figura 4 - Esempio di Report di soli totali. Una delle opzioni della struttura Report è quella che permette di stampare solo i totali e sottototali. I livelli di sottototali sono due, nel nostro caso totali per provincia e all'interno di ciascuna provincia per sesso. Alla fine viene dato il totale generale.

Struttura del database: Aiaruno.dbf					
NOME	C	12	PROVNASC	C	2
COGNOME	C	14	STIPENDIO	N	8
SEX	C	1			
DATANASC	D	8			
Gruppo/subtotale su:		PROVNASC			
Prospetto sommario (S/N)?		S	Salto pagina a gruppo/subtotale (S/N)?		
Gruppo/subtotale:		RAGGRUPPAMENTO PROVINCIA *			
Sottogruppo/sub-subtotale su:		SEX			
Sottogruppo/sub-subtotale:		SESSO			

Figura 3 - Videata comando Report. Il comando Create Report procede per videate successive nelle quali bisogna immettere i vari dati che caratterizzano la stampa. La seconda videata, quella mostrata, è quella dalla quale si definiscono le caratteristiche delle totalizzazioni.



```

* TRUNC
SET TALK OFF
P1="###,##"
P2="###,###,###"
USE ARUND
INDEX ON PROVNASC+SEX TO INUM
TOTF=0
TOTM=0
GO TOP
CLEAR
@ 1,1 SAY "TOTALIZZAZIONI PER PROVINCIA"
@ 3,1 SAY "FEMMINE" FOPRIME PERC.
MAGCHI PERC. TOT.PROV."
RIGA=5
DO WHILE .NOT. EOF()
  K1=PROVNASC
  TOTF=0
  TOTM=0
  DO WHILE K1=PROVNASC
    IF SEX="F"
      TOTF=TOTF+STIPENDIO
    ELSE
      TOTM=TOTM+STIPENDIO
    ENDIF
    SKIP
  ENDDO
  TOTF=TOTF+TOTM
  TOTGF=TOTGF+TOTF
  TOTGM=TOTGM+TOTM
  @ RIGA,1 SAY K1
  @ RIGA,8 SAY TOTF PICTURE "SP2"
  @ RIGA,21 SAY TOTF*100/TOTF PICTURE "SP2"
  @ RIGA,30 SAY TOTM PICTURE "SP2"
  @ RIGA,43 SAY TOTM*100/TOTF PICTURE "SP2"
  @ RIGA,52 SAY TOTF PICTURE "SP2"
  RIGA=RIGA+1
ENDDO
RIGA=RIGA+1
@ RIGA,1 SAY "TOTALI"
TOTG=TOTGF+TOTGM
@ RIGA,8 SAY TOTGF PICTURE "SP2"
@ RIGA,21 SAY TOTGF*100/TOTG PICTURE "SP2"
@ RIGA,30 SAY TOTGM PICTURE "SP2"
@ RIGA,43 SAY TOTGM*100/TOTG PICTURE "SP2"
@ RIGA,52 SAY TOTG PICTURE "SP2"

```

Figura 5 - Programma di Totalizzazione. Quando la struttura Report non è adatta alle nostre necessità, occorre ricorrere alla programmazione tradizionale, con tutti i problemi che questo comporta. Nel nostro caso oltre a totalizzazioni ci interessa una certa forma e un calcolo di percentuali.

l'archivio è voluminoso. Inoltre si possono eseguire, con una unica istruzione, fino a 5 calcoli. Per cui se si vogliono più somme occorre scrivere più volte il comando e il tempo globale di calcolo aumenta in proporzione.

Gli esempi mostrati in figura 2 e basati sul file visualizzato in figura 1, mostrano come si usa la funzione SUM. E cioè calcolo diretto, calcolo diretto per condizione, attribuzione del risultato ad una variabile (da usare in seguito) e infine uso del comando applicato ad una espressione matematica.

Nel nostro caso vogliamo trarre l'età media (riferita alla data di sistema indicata dalla variabile DATE) parten-

do dalla data di nascita di ciascun individuo presente nel file. Poiché la differenza tra date dà un valore numerico in giorni è possibile, dividendo per 365 ciascuna differenza ottenere un risultato in anni.

Complicando un pochino la istruzione, sempre nei limiti permessi dalla sua sintassi, si riesce ad ottenere risultati che con un linguaggio «tradizionale» in genere richiedono pesanti routine di calcolo, in particolare quando entrano in gioco date.

2) L'uso delle opzioni di totalizzazione della struttura REPORT, che abbiamo visto più volte nelle puntate precedenti, risulta molto utile quando le totalizzazioni servono solo nelle stampe e dei loro valori non è necessario conservare traccia.

In questo caso si crea il REPORT e si risponde opportunamente alle domande che fa il DB. Va capita la filosofia del comando e quindi i concetti di raggruppamento, ovvero quali campi raggruppare, e il concetto di totalizzazione ovvero su quali campi numerici totalizzare.

È possibile stampare solo i totali. In figura 3 è riportata la videata del comando nella quale bisogna indicare i campi di primo e secondo raggruppamento, e i messaggi che debbono apparire sul Report in corrispondenza degli stessi.

Nelle videate successive, nei campi numerici, occorre indicare quali campi sommare e quali no.

Il REPORT è una struttura molto potente e versatile, molto più di quello che sembra ad un suo esame sommario, per cui utilizzandola a fondo è possibile risolvere buona parte dei normali problemi che si presentano durante la manipolazione dei dati.

Ad esempio è facilmente risolvibile la mancanza di un comando COUNT, inserendo un campo numerico sempre uguale ad 1 in ogni record e considerando la relativa somma come un conteggio.

Le caratteristiche della struttura Report, che quindi permettono di capire cosa si può «fare» e cosa no con tale

struttura sono fondamentalmente due: — lavora con il concetto riga = record, per cui non è possibile, ad esempio, eseguire nell'ambito della struttura delle medie, i cui valori dipendono anche dalle righe successive; — non è molto versatile per quanto riguarda l'aspetto estetico della stampa, non è possibile per esempio comporre una stampa tipo quelle prodotte dai programmi di fatturazione (stampa della fattura) o di stipendi (stampa dei cedolini).

Ma questo è un prezzo che bisogna pagare alla facilità d'uso. In figura 4 vediamo il risultato del Report di soli totali.

3) Struttura DO... WHILE, IF ENDDIF.

Quando non è possibile o non conviene utilizzare la struttura REPORT o i comandi SUM, COUNT, è ovviamente sempre possibile ricorrere alla cara vecchia programmazione, tramite la quale con strutture «classiche» è sempre possibile raggiungere il risultato voluto.

In figura 5 vediamo un programma di totalizzazione, che lavora anch'esso sull'archivio di figura 1.

I raggruppamenti sono ottenuti indicizzando il file e usando delle strutture DO WHILE.. ENDDO, inserite l'una dentro l'altra. Con tale sistema non ci sono limiti alle possibilità di calcolo.

Nel nostro caso c'è un loop esterno:

```

DO WHILE .NOT. EOF (),
..
SKIP
ENDDO

```

con il quale si gestisce lo scorrimento di tutto l'archivio. Il loop interno è:

```

DO WHILE K1=PROVNASC
ENDDO

```

con il quale si gestisce lo scorrimento a parità di provincia di nascita. Per il calcolo dei totali per sesso, all'interno della stessa provincia, essendo i sessi due M e F, usiamo la struttura IF .. ELSE .. ENDIF.

Alla fine del loop interno si stampa-

Figura 7 - Utilizzo del comando Total. Il comando Total crea un nuovo file di struttura identica a quella del file in uso e nel quale i campi numerici sono sommati a seconda del campo di raggruppamento previsto nella sua sintassi. Il file in uso deve essere organizzato sulla stessa chiave di raggruppamento.

Figura 6 - Output del programma di Stampa. Il Report mediante programma permette una maggiore versatilità, ma comporta, ad esempio, l'inconveniente di dover gestire con apposite variabili e conteggi i salti pagina.

TOTALIZZAZIONI PER PROVINCIA					
PROV.	FEMMINE	PERC.	MASCHI	PERC.	TOT. PROV.
LI	5,200,000	39.53	7,955,000	60.47	13,155,000
MI	7,071,000	44.94	9,706,000	55.16	17,599,000
RM	2,180,000	16.91	17,001,000	89.09	19,901,000
TO	1,295,000	15.66	6,974,000	84.34	8,269,000
TOTALI	16,756,000	28.28	43,438,000	71.92	59,804,000

```

* USE ARUND
* SET TALK OFF
TOTAL ON PROVNASC TO TRUNC
24 record in totale.
3 record prodotti.
* USE TRUNC
LIST
Record: NAME          COGNOME          SEX DATEINASC PROVNASC STIPENDIO
1 LUIGIA          ROSSINI          F 17/06/55 LI          15150000
2 LUCIA          FIORINI          F 25/03/68 MI          17599000
3 LUCIA          FIORENTINI       F 13/11/61 RM          19901000
4 BEATRICE       BARONI          F 29/05/57 TO          8269200
LIST PROVNASC,STIPENDIO
Record: PROVNASC STIPENDIO
1 LI          15150000
2 MI          17599000
3 RM          19901000
4 TO          8269200

```



no i totali per provincia e si incrementano i totali generali.

Alla fine del loop esterno si stampano i totali generali, come si può vedere nella figura 6 che rappresenta l'output del programma.

4) Comando TOTAL

Il comando TOTAL è un comando molto evoluto che risulta particolarmente utile quando dei nostri totali occorre eseguire ulteriori e complesse elaborazioni.

Il comando TOTAL crea un nuovo file di totali, di struttura identica a quello base, i cui campi numerici contengono la somma dei campi numerici dei singoli record, mentre i campi non numerici vengono riempiti con i campi corrispondenti del primo record che rispetta la condizione di raggruppamento e quindi in pratica non conservano alcun interesse.

Le condizioni da rispettare sono il fatto che l'archivio base deve essere indicizzato secondo la chiave di raggruppamento dei totali, e poi che le dimensioni dei campi numerici siano tali da permettere il caricamento dei dati numerici in arrivo.

Il file dei totali creato è un file tipo \*. DBF a tutti gli effetti può essere quindi comunque manipolato per usi successivi.

In figura 7 vediamo l'utilizzo in comandi diretti dell'istruzione TOTAL eseguita sul nostro file indicizzato per provincia. Costruito il file di totali lo si attiva con il comando USE (nome file) e se ne eseguono due listati.

Nel primo si vede come i campi di tipo non numerico e non di raggruppamento perdano di valore, nel secondo vengono listati solo i campi di interesse.

**Modifica della struttura e del contenuto di archivi**

Altra problematica classica è quella relativa alla manipolazione di archivi, intesa questa volta come modifica sia della struttura sia del contenuto degli archivi.

È possibile «rivoltare come un pedolino» gli archivi (a Roma si dice così) con pochi comandi, ben assestati. L'unica precauzione da prendere è quella di fare delle copie di sicurezza degli archivi iniziali e intermedi, perché in caso di errore nella impostazione dei comandi, si corre durante l'operazione il rischio di perdere dei dati.

Anche in questo caso vanno distinti i due ambiti di lavoro. In comandi diretti dove si può lavorare con le funzionalità interattive di CREATE, MODIFY, COPY e in programmazione, dove tali funzionalità o non si possono neanche attivare o dove risultano troppo pericolose da far maneggiare ad un operatore inesperto.

Figura 8 - Modifica della struttura di un file. Viene mostrato il comando per mezzo del quale è possibile eseguire da programma la modifica della struttura di un file. In pratica la struttura diventa a sua volta un file sul quale è possibile eseguire facilmente aggiornamenti.

```
* MDUNO
* MODIFICA DI UNA STRUTTURA IN PROGRAMMA
USE ARUNO
COPY STRUCTURE EXTENDED TO STRUTT
USE STRUTT
LOCATE FOR FIELD_NAME="COGNOME"
REPLACE FIELD_LEN WITH 20
APPEND BLANK
REPLACE FIELD_NAME WITH "ETA"
REPLACE FIELD_TYPE WITH "N"
REPLACE FIELD_LEN WITH 2
REPLACE FIELD_DEC WITH 0
USE
CREATE ARDUE FROM STRUTT
APPE FROM ARUNO
REPLACE ALL ETA WITH YEAR( DATE() )-YEAR( DATANASC )
LIST
```

1) Modifiche di struttura e di contenuto eseguite in comandi diretti.

Quella di dover modificare la struttura di un file è una eventualità ricorrente sia quando ci si è dimenticati di inserire campi, sia quando ci si accorge che le dimensioni di un campo sono insufficienti, sia quando occorre aumentare il contenuto di informazioni del file.

Il problema consiste nell'eseguire l'operazione nel minor tempo possibile e «perdendo» meno dati possibili.

Il DB III offre numerose possibilità per raggiungere il risultato voluto, anche in questo caso quindi sta all'abilità e alla sensibilità dell'operatore la scelta di una strada o di un'altra, scelta che deve sempre tener conto di fattori di sicurezza.

La strada più sicura per modificare la struttura di un file pieno di dati, che è quella che si seguiva con il DB II, e

che consiste nell'utilizzare il comando:

COPY STRUCTURE TO <nome file destinazione>

con il quale partendo dal file originale si ottiene un file di struttura identica, ma vuoto di dati. Poi ci si sposta sul file vuoto e si esegue un comando:

MODIFY STRUCTURE

con il quale si eseguono le modifiche strutturali senza pericoli sui dati. Infine rimanendo sul nuovo file vuoto, ma con la struttura modificata si esegue il comando:

APPEND FROM <(nome file origine)>

Con questo sistema il file origine viene manipolato solo in lettura e quindi anche in caso di errori o problemi rimane integro. Il comando APPEND funziona a parità di nome di campo e non di tipo o di lunghezza,

Record	NO ME	COGNOME	SEX	DATANASC	PROVNASC	STIPENDIO	ETA
1	LUCIA	GIROENTINI	F	13/11/61	RM	1090000	19
2	DENIAMINO	GIALLI	M	21/05/45	LI	1075000	35
3	NATALE	ARANCIO	M	04/07/58	LI	1450000	22
4	LIDIO	BIANCHINI	M	27/05/51	LI	1170000	29
5	GIUSEPPE	AGATINI	M	04/04/55	RM	6650000	25
6	MARIO	LUSSU	M	22/03/65	TO	540700	15
7	LUDOVICO	BIANCHI	M	19/11/54	LI	2190000	26
8	CHIARA	ROSA	F	25/03/68	MI	1456000	20
9	DORINA	MALTESE	F	11/11/65	MI	4560000	15
10	CARLOTTA	LUDOVIGI	F	22/11/50	MI	1075000	17
11	ALDO	AZZURRI	M	15/02/56	MI	2009000	24
12	WALTER	CHAIRINI	M	25/05/61	RM	3456000	19
13	ALESSANDRO	MALINI	M	12/12/56	MI	2009000	24
14	DONATELLO	DOMINI	M	21/01/74	TO	6433300	46
15	LUISA	ROGGINI	F	19/06/55	LI	5200000	25
16	MARGHERITA	VIOLA	F	05/10/54	RM	1070000	26
17	LUIGI	ROSSI	M	09/06/58	RM	1200000	22
18	WALTER	NERI	M	11/01/55	RM	1100000	25
19	BEATRICE	MARRONI	F	29/06/57	TO	1295000	23
20	WALTER	LUCIOLI	M	11/02/55	RM	4100000	25
21	BIAGIO	MALETTI	M	19/02/57	RM	1295000	23
22	MARCO	VERDI	M	20/03/64	LI	1250000	16
23	MARIO	BRUNO	M	20/12/62	MI	4745000	18
24	CARLO	BINI	M	12/12/61	MI	1345000	19

Figura 9 - Out del programma di Modifica Struttura. Va notata la possibilità dell'istruzione REPLACE per mezzo della quale, nel nostro caso, «riempiamo» il campo ETA di tutti i record con il risultato di un calcolo eseguito sul campo DATANASC.



per cui i dati vengono copiati il più possibile.

Ad esempio se un campo carattere cambia dimensione l'APPEND trasferisce i contenuti troncando o inserendo blank.

Nel caso invece che la modifica comporti anche modifiche dei nomi dei campi l'APPEND non è in grado di riconoscere (ovviamente) la relazione tra i campi. In tale caso occorre fare due passaggi, utilizzando un file intermedio che contenga entrambi i campi (quello con il vecchio nome e quello con il nuovo) e poi con un comando di REPLACE si ottiene il passaggio:

```
REPLACE ALL <nuovo campo> WITH
<vecchio campo>
```

ghezza del campo COGNOME e l'inserimento di un campo ETÀ, che viene riempito eseguendo un REPLACE del campo ETÀ con il risultato di una elaborazione del campo DATANASC (in figura 9 il risultato).

### Il comando UPDATE

Altro comando avanzato, ma di uso diffusissimo è l'UPDATE, con il quale si ottiene l'aggiornamento di uno o più campi di un archivio prelevandoli da un altro file.

Ad esempio un file articoli di un commerciante nel quale sia presente un campo prezzo da parte del fornitore. Se il fornitore aggiorna i propri prezzi, il commerciante deve aggiornare

*Figura 10 - Comando Update. Aggiornamento ottenuto trasferendo dati da un archivio origine ad un archivio destinazione. La condizione da rispettare è che i due archivi siano organizzati con la stessa chiave, che in pratica mette in relazione i due archivi.*

```
* ARJND
* AGGIORNAMENTO DI UN CAMPO DI UN FILE.
USE ARJND
INDEX ON COGNOME TO INJND
LIST COGNOME,STIPENDIO
SELECT 2
USE ARDUE INDEX INJUE
SELECT 1
UPDATE ON COGNOME FROM ARDUE REPLACE STIPENDIO WITH B->STIP
LIST COGNOME,STIPENDIO
```

Ripetiamo, le possibilità sono numerosissime e quindi si riesce ad eseguire la trasformazione perdendo meno dati possibili.

2) Modifica di strutture da programma.

In applicazioni particolarmente avanzate può risultare necessario modificare la struttura di un file all'interno di un programma applicativo, dove non si possono utilizzare i comandi diretti ora visti.

Senza entrare nei dettagli diremo che esiste un comando che permette di «tradurre» la struttura di un archivio in un file in cui i record sono i campi del file origine. I campi di tale particolarissimo file sono:

```
FIELD_NAME
FIELD_TYPE
FIELD_LEN
FIELD_DEC
```

e il comando che esegue la trasformazione è:

```
COPY STRUCTURE EXTENDED TO
<nome file struttura>
```

mentre il comando inverso, per mezzo del quale dal file struttura si riottiene il file dati modificato è:

```
CREATE <nome file dati> FROM <nome file struttura>
```

Quindi la modifica della struttura diventa un qualsiasi aggiornamento di file sempre eseguibile da programma. In figura 8 vediamo un programma per mezzo del quale il nostro archivio di prova subisce la modifica della lun-

re il relativo campo del proprio archivio.

Se il fornitore gli invia un dischetto con un file con codice articolo e nuovo prezzo, il commerciante può evitare di digitare tutti i nuovi prezzi eseguendo l'update (espressione che ormai anche nel linguaggio corrente significa aggiornamento).

Il comando UPDATE lavora con i due file in uso e pretende che i due file siano organizzati a parità di chiave, la sua sintassi si può vedere nella figura 10, nella quale viene eseguito l'aggiornamento del campo stipendio del file di prova, con i dati provenienti da un archivio esterno.

### SET RELATION

Abbiamo più volte visto nelle puntate precedenti la funzionalità SET RELATION, che permette di prelevare dati là dove servono in maniera indipendente dall'organizzazione fisica dei file.

Vogliamo esemplificare ancora questa funzionalità sulla quale si basa la superiorità di un Data Base Management System di tipo relazionale rispetto ad un gestore tradizionale di archivi. Abbiamo tre archivi, il primo è il nostro archivio di prova, poi abbiamo una tabella di province, comprendente tre campi:

```
SIGLA PROVINCIA
NOME DELLA CITTÀ
CODICE REGIONE
```

Il terzo archivio è una tabella di regioni in cui al codice regione corrisponde il suo nome.

Il problema è quello di eseguire una stampa dove appaiono:

```
SIGLA PROVINCIA (presente nell'archivio 1)
NOME DELLA CITTÀ (presente nell'archivio 2)
NOME DELLA REGIONE (presente nell'archivio 3)
```

La sigla della provincia presente nell'archivio 1 è relazionata alla sigla provincia chiave dell'archivio 2. Mentre la sigla della regione, presente come campo nell'archivio è relazionata con la chiave dell'archivio 3.

In figura 11, 12 e 13 vediamo sia il programma che genera questa «catena» di relazioni, sia il risultato pratico sia lo STATUS, ovvero la situazione di archivi e indici aperti e di relazioni definite che permettono l'operazione.

Dalle figure non si può verificare, ma la stampa è istantanea, ovvero non subisce perditempi dovuti al fatto che in pratica i dati provengono da svariati archivi. Si può inoltre verificare il fatto che, se il codice messo in relazione non trova corrispondenze nell'archivio relazionato non si attiva una condizione di errore, il che permette di usare con più tranquillità tali comandi.

La filosofia di tutto ciò consiste nel riuscire realmente a svincolare gli ambiti applicativi dall'organizzazione fisica dei dati, per cui questi, per mezzo di semplici, ma potenti comandi assumono un certo aspetto logico funzionale all'uso di quel dato momento, ma del tutto indipendente da quello fisico, che al limite non deve essere conosciuto, anzi che è bene che non sia conosciuto.

### Conclusioni

Alla fine di un corso occorrerebbe fare... gli esami (niente paura non abbiamo nessuna intenzione di interrogarvi), mentre le conclusioni in genere si traggono alla fine della prova di un prodotto.

Vogliamo comunque fare alcune valutazioni e considerazioni finali, che spieghino il fenomeno DB e che possano risultare utili a chi decide di investire il proprio tempo nell'apprendimento di un linguaggio di programmazione.

### Vantaggi del DB III

Il primo grande vantaggio è che il DB III è ormai lo standard tra i package Data Base Management System su PC. Questo vuol dire che chi impara a usare e a programmare in DB III impara qualcosa di «vendibile sul mercato» molto più ad esempio di quanto non sia vendibile una cultura su un



```

. TYPE REIND.PRG TO PRINT
* REIND
* RELAZIONE DOPPIA
SELE 1
USE ARUNO
SELE 2
USE DUE INDE DUE
LIST
SELE 3
USE TRE INDE TRE
LIST
SELE 2
SET RELA TO DUE3 INTO TRE
SELE 1
SET RELA TO PROVNASC INTO DUE
LIST COGNOME,PROVNASC,B->DUE2,C->TRE2 FOR SCX="F"

```

linguaggio poco usato oppure di quanto non sia vendibile un prodotto software che deve entrare in concorrenza con migliaia di prodotti analoghi.

Un altro grande vantaggio del DB III è il suo basso livello di accesso, ovvero è facile da usare sin dall'inizio, e via via che lo si usa in maniera più massiccia adegua la sua potenzialità alle differenti esigenze.

Questo agevola anche il «non esperto» al quale per incominciare a usare il DB III non è richiesta alcuna conoscenza di concetti di informatica e che nella sua crescita culturale trova uno strumento che gli permette di mettere correttamente a fuoco le varie problematiche di trattamento dati cui progressivamente si trova di fronte.

Il terzo vantaggio del DB III è la sua diffusione che sta provocando una vera e propria «ricaduta di prodotti ausiliari» che migliorano questa o quella funzionalità o che agevolano l'uso del prodotto. Questo è notoriamente indice di successo ed è un elemento che in sede di decisione deve avere il suo peso.

Per fare un esempio, esiste un prodotto, il Quickindex della Fox & Geller's, la cui unica funzione è quella di velocizzare il comando INDEX del DB III, che è notoriamente uno dei comandi più critici. I tempi si riducono a circa un terzo se il Quickindex viene usato da DB III, o a un decimo se usato da DOS.

Se tra gli utilizzatori nasce una esigenza è il mercato stesso che risponde immediatamente mettendo a disposizione prodotti che la risolvono.

Se questo vale per i produttori esteri, vale a maggior ragione per la Ashton Tate che produce il DB III, per il quale è già uscito in America, una nuova release (chiamata DB III Plus), che ci ripromettiamo di presentarvi al più presto nella versione italiana, curata dal distributore italiano EIS.

### Problemi che si incontrano durante l'uso del DB III

Il DB III non è comunque esente da critiche, ha dei limiti e in certi casi dei comportamenti al limite del «bug»,

Figure 11, 12 e 13 - Comando Set Relation. Mostriamo un uso avanzato del SET RELATION. Viene creata «una catena» di relazioni tra tre archivi, in modo che in out vadano dati presi dai tre archivi. Il programma è in figura 11, il suo out in figura 12, mentre in figura 13 mostriamo lo STATUS che specifica gli archivi, gli indici e le relazioni che devono essere attivi nel momento in cui si esegue il comando.

che qui vogliamo accennare, anche perché nella prossima versione Plus saranno in buona parte risolti.

Manca una gestione degli errori, e in certi casi gli errori sono fatali, e questo in un linguaggio destinato ad essere utilizzato per applicazioni professionali è grave.

La specifica Picture, che come noto permette di dare un certo formato estetico agli output, non è utilizzabile nelle strutture REPORT, e se pur utilizzabile produce strani effetti con il comando di input GET.

Questo è un limite piuttosto sentito in Italia dove qualsiasi calcolo in lire deve prevedere numerose cifre per cui è indispensabile ricorrere alle virgole come separatori di migliaia.

È possibile eseguire un indice con un campo data, in tal caso i record vengono messi correttamente in ordine cronologico. Purtroppo non è possibile creare indici composti con campi di tipo differente, occorre tradurre tutti i campi non di tipo stringa in stringa, e questo comporta la perdita dell'ordine cronologico, in quanto le date vengono messe questa volta in ordine alfabetico.

La relazionalità tra gli archivi presenta dei limiti in quanto mentre è possibile mettere un primo archivio in relazione con un secondo, questo con un terzo e così via, non è possibile mettere in relazione due campi di un archivio ciascuno in relazione con un altro.

Questi problemi elencati sono tutti ovviabili specie in programmazione,

```

. DO REIND
Record DUE1 DUE2 DUE3
1 AL ALESSANDRI 01
2 MI MILANO 02
3 TO TORINO 01
Record TRE1 TRE2
1 01 PEMONTE
2 02 LOMBARDIA
Record COGNOME PROVNASC B->DUE2 C->TRE2
1 FIORENTINI RM MILANO LOMBARDIA
8 ROSA MI MILANO LOMBARDIA
9 MALTESI MI MILANO LOMBARDIA
10 LUDOVIGI MI MILANO LOMBARDIA
15 ROSSINI LI
16 VIOLA RM
19 MARRONI TO TORINO PEMONTE

```

```

. display status
Database selezionato:
Area di lavoro: 1, database attivo: A:ARUNO.dbf, alias: ARUNO
In relazione con: DUE
Relazione: PROVNASC
Area di lavoro: 2, database attivo: A:DUE.dbf, alias: DUE
File indice: A:DUE.ndx, chiave: DUE1
In relazione con: TRE
Relazione: DUE3
Area di lavoro: 3, database attivo: A:TRE.dbf, alias: TRE
File indice: A:TRE.ndx, chiave: TRE1
Premere un tasto qualsiasi per continuare.

```

dove è possibile comunque trovare soluzioni alternative.

Altro aspetto da tenere in considerazione sono le prestazioni del DB III, in termini di tempo, da prevedere in applicazioni pesanti (più archivi di migliaia di record).

Le prestazioni dipendono innanzitutto dal tipo di macchina in quanto possono variare di un ordine di grandezza se si passa ad esempio, da un PC normale a floppy a un PC AT con hard disk.

Le varie funzionalità della procedura non vengono penalizzate neanche in caso di archivi pesanti se seguono la logica di organizzazione degli archivi. Ad esempio il tempo per eseguire una FIND su un archivio indicizzato è lo stesso.

Vengono penalizzate quelle funzionalità che non seguono l'organizzazione logica degli archivi, ad esempio una ricerca sequenziale (LOCATE FOR condizione) comunque scorra l'archivio, oppure un INDEX creato per un uso estemporaneo.

Queste sono funzionalità definibili BATCH, cioè non interattive, e che possono essere previste solo valutando i tempi di esecuzione, e quindi programmando l'esecuzione (ad esempio di notte).

Ma anche questo aspetto troverà soluzioni soprattutto con il DB Plus (indici 10 volte più veloci e sort 2 volte) e con le nuove serie di prodotti ausiliari.

Contiamo di presentarvi al più presto le ultime novità e quindi di riprendere il discorso DB III.



# MASTER NETWORK

Importanti novità nel mercato dell'informatica!

## ACQUISTARE HARDWARE E SOFTWARE A PREZZO DI RIVENDITORE PIÙ IL 10%.

La MASTER-NETWORK è una organizzazione formata da utenti di computers, quindi anche da Te!

Nostro scopo è quello di diffondere concretamente l'informatizzazione su vasta scala, effettuando una politica sana ed estremamente efficace: L'acquisto di merci sui mercati più favorevoli nella giusta quantità ed al giusto prezzo.

### ECCO ALCUNI ARTICOLI AL NOSTRO COSTO D'ACQUISTO:

MODEMPHONE ACC 303



Modem a 300 BAUD da collegare con RS232 CARD con telefono provvisto di 10 memorie.

**L. 220.000**

MANNESMANN TALLY MT80 PC



Stampante a 80 colonne 130 CPS, interfaccia parallela, grafica IBM.

**L. 460.000**

Scheda monocromatica grafica tipo: HERCULES II



Risoluzione 720 HX 348W

**L. 210.000**

Scheda di espansione 512 K a 0 RAM.



**L. 110.000**

HARD DISK



Completati di controller, cavi, ed istruzioni. Marche: SHUGART, TANDON, SEAGATE.

10MB

**L. 1.100.000**

20MB

**L. 1.350.000**

**Framework L. 1.000.000**

**Gestione Aziendale**

**L. 1.000.000**

**Dos IBM 3,1 L. 100.000**

**Software in distribuzione esclusiva**

per COMMODORE, SINCLAIR, APPLE,

Monitor HANTAREX



Boxer 12 PC 12 Pollici monocromatico, sfostori verdi, schermo antiriflessi

**L. 170.000**

ABBONATI quindi, ed avrai diritto a comperare tutti gli articoli del nostro magazzino con la sola maggiorazione del **10%** sul nostro costo di acquisto. Riceverai altresì la tessera personale, con la quale potrai partecipare a tutte le nostre azioni promozionali. Contestualmente, ti sarà

inviato il ns. ricco catalogo contenente tutti gli articoli e periodicamente, ti saranno inoltre inviati gli aggiornamenti sulle ultime novità di hardware e software e le speciali proposte sulle giacenze di magazzino.

L'abbonamento non ti impegna ad alcun acquisto.

Le vendite sono effettuate per corrispondenza garantendo l'invio entro 48 ore dalla ricezione dell'ordine.

I costi delle merci sono I.V.A. esclusa.

Tutte le merci vendute sono con garanzia.

SS/DD

**L. 2.000 L. 2.000 L. 2.000**

DISKETTES



**VEREX  
VERBATIM**



calcolatrice

8 cifre

12 funzioni  
£ 4000

DS/DD

**L. 2.500 L. 2.500 L. 2.500**

Favolose promozioni per COMMODORE, COMMODORE AMIGA, ATARI, SINCLAIR, APPLE, EPSON, IBM, ecc.

### PERSONAL COMPUTER XT COMPATIBILE



Composto da: 256 K, 2 Drives, Tastiera U.S., Scheda grafica colore o monocrome, 1 Monitor 12 pollici f.v.

**L. 1.690.000.**

Desidero sottoscrivere un abbonamento al servizio MASTER NETWORK ALLE CONDIZIONI SOTTODESCRITTE: //

- 1) Pagherò L. 20.000 in contrassegno al ricevimento del 1° catalogo e della tessera personale di abbonamento.
- 2) Potrò pertanto acquistare le merci in catalogo con il solo aumento del 10% sul vostro costo di acquisto.
- 3) L'abbonamento non comporta alcun impegno di acquisto da parte mia di qualsiasi natura o genere. Otterrò comunque per la durata di un anno i Vs. aggiornamenti.

COGNOME ..... NOME .....

INDIRIZZO .....

PROFESSIONE .....

CAP ..... CITTA ..... PROVINCIA .....

FIRMA .....

Spedisci l'allegato Coupon presso i nostri recapiti postali di:

20123MILANO - Via Monti, 8

00198 ROMA - Via Savoia, 78

80144 NAPOLI - Via E. Gianturco, 50/A

Telex 316329 EXECMI

Telex 613458 EXECRO

Telex 722039 DBASI