



di Francesco Petroni

Creazione e gestione dei file dati

Seconda parte

La prima parte del corso, pubblicata sullo scorso numero di MC, aveva una finalità introduttiva e quindi in pratica è servita per far capire la filosofia e le modalità di approccio allo strumento DB.

Con la seconda parte si entra nel vivo del corso e si comincia a parlare di archivi. Si vedrà come costruire un archivio, come lo si gestisce, come si manipolano i dati in esso contenuti, ecc.

Propedeutica a questa parte del corso è la conoscenza di alcuni concetti fondamentali come il concetto di File, il concetto di Struttura, il concetto di Campo.

Concetti fondamentali struttura di un File, Campi, Record

Il file (o archivio) va inteso come un insieme logicamente composto di informazioni. Nel file vanno distinti gli elementi che costituiscono il contenitore (Struttura, Campi, tipo e lunghezza dei Campi, Controlli dei Campi, ecc.) dal contenuto.

Il contenuto è un insieme di Record, e per Record si intende un insieme, di dimensioni costanti e predeterminate all'atto della creazione del file,

di informazioni tra loro rigidamente correlate. Il Record è il minimo elemento dell'archivio, anzi la quantità di informazioni contenute nell'archivio si misura in numero di Record.

Come detto ogni Record è un insieme di informazioni, ciascuna delle quali è associata ad un campo che ne determina e specifica il significato.

La rappresentazione grafica più immediata della struttura di un File è costituita da una tabella rettangolare, in cui la dimensione orizzontale è fissa, in quanto comprende le specifiche della struttura e cioè nome dei campi, tipo, lunghezza, eventuali controlli (vedi tabella di fig. 1).

Invece la dimensione verticale dipende dal numero di record inseriti ed è quindi generalmente variabile, in quanto permette accodamenti, cancellazioni, inserimenti, e riorganizzazioni. Tutte queste operazioni debbono avvenire senza che la struttura (la parte orizzontale della tabella) vari.

L'argomento della seconda puntata del corso riguarda la struttura (ovvero come si crea o si modifica una struttura) e l'organizzazione in Record, e cioè come, una volta definita la struttura si possono inserire, ricercare, mo-

dificare Record e, trattandosi di un corso DB II e DB III, quali istruzioni o modalità di lavoro caratterizzano questi prodotti.

Creazione di una struttura

Per creare la struttura di un File si utilizza il comando CREATE «nome del file». Sia il DB II che il DB III rispondono mostrando una tabellina per l'immissione dei dati che caratterizzano la struttura e cioè.

Nome del campo	
Tipo del campo	carattere data (solo DB III) numerico logico memo (solo DB III)
Lunghezza del campo	in caratteri
Numero dei decimali	in caratteri

La lunghezza può arrivare fino a 256 per i Campi carattere (noi diremmo alfanumerici), è fissa in 8 per i Campi Data, è fissa in 1 per i Campi logici, che assumono solo i due valori T (true = vero) e F (false = falso).

I Campi numerici possono arrivare fino a 15 caratteri, ma la loro reale lunghezza dipende dalla precisione di calcolo permessa dal linguaggio (10 cifre per il DB II, 15 per il DB III), mentre la parte decimale va calcolata col metodo classico di tutti i linguaggi. In pratica per rappresentare il numero 99,999 è necessario un Campo Numerico lungo 6 di cui 3 decimali.

Il campo Memo (vale solo per il DB III) occupa una lunghezza fissa di 10 caratteri, che in pratica costituiscono la chiave di accesso ad un altro file (tipo *.DBT nella tipologia illustrata nella prima puntata) e gestito totalmente dal DB III.

In DB II la struttura può compren-

NUMERO RECORD	CAMPI (tipo)					
	(car.)	(car.)	(f)	(data)	(numerico)	
RECNO()	NOME	COGNOME	SEX	DATA NASC.	PROV.	STIPENDIO
1	LUIGI	ROSSI	M	09-Jun-58	RM	1,200,000
2	MARCO	VERDI	M	20-Mar-64	PA	1,250,000
3	LUDOVICO	BIANCHI	M	29-May-51	LI	2,190,000
4	CARLOTTA	GIALLI	F	21-May-45	LI	1,875,000
5	MARGHERITA	VIOLA	F	03-Oct-54	RM	1,090,200
6	WALTER	NERI	M	11-Jan-55	RM	1,100,000
7	BENIAMINO	MARRONI	M	29-Jun-57	RM	1,295,000
8	GIUSEPPE	ARANCIO	M	04-Sep-58	LI	1,450,000
9	MARIO	BRUNO	M	20-Dec-62	TO	1,345,000
10	CHIARA	ROSA	F	25-Mar-60	MI	1,456,000
11	ALDO	AZZURRI	M	15-Feb-56	MI	2,009,000
...						
...						

Figura 1 - Rappresentazione Grafica di un Archivio. La struttura è il contenitore costituito da nome, tipo e dimensioni dei campi. Il contenuto è formato da record. Ciascun record è un insieme di informazioni il cui significato reale dipende dalla struttura nel quale sono immessi. Ad esempio il dato «ROSA» è un cognome in quanto è inserito nel campo cognome.

dere fino a 32 campi, che in applicazioni di una certa complessità sono indubbiamente insufficienti, e l'occupazione totale della struttura può arrivare fino a 1500 byte.

Il DB III, oltre ad avere i due tipi di campo in più, ne permette fino a 128 per una occupazione totale di 4000 byte al massimo, che in pratica aumenta nel caso si utilizzino campi Memo.

Il campo Data, del DB III, non è né del tipo carattere né del tipo numerico, ma accetta calcoli numerici (differenza tra date, ed il risultato è numero di giorni, somma tra una data e un numero di giorni che dà per risultato una nuova data).

remo un piccolo programma di immissione dati con alcuni controlli sui dati immessi. Ad esempio, se in una struttura abbiamo inserito un campo Codice Provincia, di tipo carattere e di lunghezza 2 caratteri, immettendo i dati in modalità comandi diretti il DB controlla solo tipo e lunghezza. Invece un controllo, eseguibile solo da programma, è quello di verificare l'esistenza della sigla immessa in una tabella, al limite, in un altro archivio.

Altro esempio: in un archivio che contiene informazioni sul personale di una certa ditta esistono in ogni record, riferito a ciascun dipendente della ditta, due date, quella di nascita e quella

il record successivo pronto per l'immissione, e così via. Per interromperla basta dare un RETURN a vuoto all'inizio del nuovo Record.

Come ci si muove all'interno di un file

Il comando USE <nome del file> apre il file e posiziona il numero del record a 1. Se non si sta utilizzando alcun file indice, l'ordine in cui sono logicamente messi i dati è fornito dal loro numero record, ovvero sono messi in ordine di immissione.

Dopo l'USE sono dunque attivi i

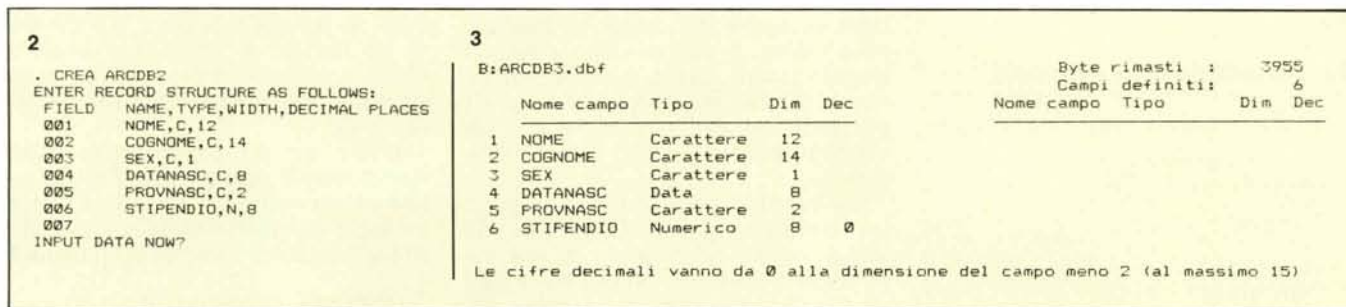


Figura 2 e 3 - Comando CREATE <nome del file>. Il comando fa apparire una maschera in cui, in maniera assolutamente intuitiva, vanno immesse le informazioni necessarie alla definizione della struttura del nuovo file. Il DB3 esegue anche un conteggio dei byte consumati.

Può essere manipolato per mezzo di funzioni specifiche (tipo mese, anno, giorno e funzioni che lo trasformano in stringa e viceversa). Sono tutte possibilità che nelle procedure che si basano sul calcolo di date, come tutte le procedure di tipo amministrativo-gestionale, risultano utilissime.

In figura 2 e 3 vediamo le maschere di immissione di una struttura presentate dal comando CREATE <nome del file> rispettivamente in DB II e in DB III.

Esiste ovviamente un comando (DISPLAY STRUCTURE) che permette di vedere una struttura appena creata (nella configurazione di partenza a tale funzione è destinato il tasto F6), in quanto quando si lavora con un file, qualsiasi cosa si intenda farne, bisogna conoscerne la struttura.

Una volta definita la struttura è possibile cominciare ad inserire dati. Le modalità di immissione sono due. La più semplice è quella permessa dal lavoro in comandi diretti, la seconda è quella che consiste nel far gestire tutte le operazioni sul file da una apposita procedura (ovvero da un insieme di programmi).

La differenza tra le due modalità è evidente: con la prima si è in grado di eseguire sui dati immessi solo i controlli automatici (fatti dal DB) relativi a quel tipo campo, con la seconda tutti i dati immessi possono essere inserite in routine, anche complesse, di validazione. Alla fine della puntata presente-

di assunzione. Immettendo i dati direttamente il DB esegue solo il controllo formale del dato, che cioè quella data esista, il controllo del rapporto logico tra le due date può essere fatto solo da programma.

Gestione di un archivio in comandi diretti

Per immettere i dati in un archivio la prima cosa da fare è richiamarlo tramite il comando USE <nome archivio>. A questo punto l'archivio è a disposizione e cioè ci si può lavorare sopra e sono attivi i vari comandi che ne permettono la gestione. Ad esempio: DISP STRU mostra la struttura e indica il numero di record LIST [varie opzioni] ne visualizza il contenuto

Se dobbiamo immettere dati il comando è APPEND, che conviene tradurre in accoda. Infatti ogni nuovo record viene accodato e il DB gli attribuisce un numero che lo identificherà in seguito, e che può essere visto come un campo che contiene il progressivo del record da chi usa l'archivio.

Il comando APPEND genera un nuovo record (vedi fig. 4), il cui numero appare il alto, e presenta una maschera per immettere i dati. La maschera è composta dal nome del campo e, in inverso, da uno spazio di lunghezza pari al campo da immettere.

Finita l'immissione dei dati appare

nomi dei campi ed è attivo il primo record. I comandi utilizzabili sono: DISP che visualizza il record attivo oppure ?[RECNO()], <nome del campo>, <nome del campo> DB III ? #, <nome del campo>, <nome del campo> DB II

Il punto interrogativo è il comando di PRINT che si può riferire sia a variabili sia ai nomi dei campi del file in uso in quel momento, per cui una volta posizionatisi su un record (nel nostro caso il primo) se ne può visualizzare il contenuto o parte del contenuto.

Per muoversi all'interno del file per ricercare un record specifico il modo più semplice è conoscerne il numero record e il comando di movimento è GO <numero del record> altri comandi di movimento sono skip <+/-numero> che muove il puntatore avanti ed indietro i comandi di inizio e fine file (associabili a GO)

BOF(), EOF() variabili logiche.

Raggiunto il record che interessa, si può visualizzare con il DISP o con le varie opzioni viste prima.

Per modificare un record in comandi diretti il comando da usare è EDIT, che visualizza la stessa maschera dell'append ma con i campi «pieni». Il comando EDIT può essere usato anche passando direttamente il numero del record (es. EDIT 56).

Quando un record viene immesso il DB gli attribuisce un numero, che altri

non è che il numero progressivo di immissione. Tale numero può essere visualizzato e può anche essere utilizzato per richiamare il record, ammesso di conoscerlo.

Il numero del record (RECNO() in DB3 oppure # in DB2) perde molta importanza quando si lavora con un file indice. Il file indice comporta un diverso modo di sequenziare i record e quindi un diverso modo di movimento all'interno del file e un diverso modo di ricerca.

In ogni caso è indispensabile saperli muovere bene all'interno di un file e quindi vi suggeriamo di fare «ginnastica». In figura 5 c'è un output di un esercizio di movimento su e giù per il file.

La cancellazione di un record

Il DB consente una cancellazione

Record n.	28	
NOME	:	:
COGNOME	:	:
SEX	:	:
DATANASC	:	/ / :
PROVNASC	:	:
STIPENDIO	:	:

Figura 4 - Il comando APPEND costituisce la maniera più semplice per inserire dati in un archivio. Basta aprire un archivio e impostare APPEND che appare una maschera totalmente confezionata dal DB, in cui appaiono sulla sinistra i nomi dei campi e sulla destra, in negativo, le zone per l'immissione dei dati, già dimensionate in funzione della lunghezza del campo. In alto appare il numero del record che si sta accodando.

dei record a due livelli, la cancellazione logica e la cancellazione fisica.

Con quella logica il record continua ad esistere nel file, conserva cioè il suo numero record, ma gli viene attribuito un asterisco che indica che è stato cancellato.

Il comando è DELETE, che usato «liscio» opera sul record attivo. In realtà consente una sintassi più complessa che permette di cancellare più record contemporaneamente:

DELETE <intervallo> <condizione>

Una volta cancellati logicamente dei record possono essere considerati o meno, ad esempio in un calcolo o in una visualizzazione, tramite un comando di SET.

SET DELETED on/off

I record cancellati logicamente possono essere riattivati con il comando RECALL che ha una sintassi identica al suo contrario DELETE.

La cancellazione fisica avviene con il comando PACK, che provoca la riassegnazione dei numeri record. Ovvero se su un file di 100 record cancelliamo il 50mo, dal 51mo in poi il numero record viene riattribuito.

Il PACK è una operazione quindi di tipo «pesante» che va eseguita con

certe precauzioni. In generale conviene lavorare con il DELETE, e, saltuariamente, con operazioni di tipo BATCH, eseguire il PACK. Mai e poi mai il PACK può essere immesso in una procedura di gestione maschere.

Il comando REPLACE

Esiste un altro comando che permette di modificare il contenuto di uno o più campi di uno o più record, è il comando REPLACE la cui sintassi è:

REPLACE <campi> WITH <espressione> FOR/WHILE <condizione>

Il Replace è quindi un comando di assegnazione che permette di far assumere ad uno o più campi un determinato valore. E questo vale sia per un singolo record, quello su cui è posizionato il puntatore in quel momento, sia per tutti i record che rispettino una certa condizione e tutti i record in assoluto.

Ad esempio, se si vuol assegnare ad un campo chiamato Progressivo lo stesso valore del numero record aumentato di 1000 e al campo Nominativo il valore del campo Cognome più quello Nome, ambedue trasformati in caratteri maiuscoli, l'intera operazione può essere fatta con un unico comando Replace

REPLACE Progressivo WITH Recno() + 1000, Nominativo WITH UPPER (Cognome) + UPPER (Nome)

È evidente che se l'operazione riguarda più record di un File consistente l'operazione di Replace richiede un certo tempo.

Un'altra modalità di utilizzo del comando REPLACE è quella costituita da una maschera di immissione dati, per la quale può essere meno rischioso lavorare su variabili di comodo, sulle quali è possibile eseguire controlli e calcoli a volontà. Solo alla fine, quando i dati siano stati tutti validati, si inseriscono nel record tramite un'unica operazione di APPEND seguita immediatamente da un REPLACE.

La maggior sicurezza consiste nel fatto che l'APPEND è immediatamente seguito da un REPLACE e quindi viene minimizzato l'intervallo di tempo tra la predisposizione del nuovo record e l'effettivo riempimento dei suoi campi.

Al contrario, se si lavora direttamente sui campi occorre dare subito il comando di APPEND, per cui se succede qualcosa che interrompe il lavoro (ad esempio un errore nel programma o una caduta di tensione) c'è il rischio che il record sia stato costruito, ma che i dati non siano stati immessi o lo siano stati in modo errato.

L'inconveniente più grave è costituito dal fatto che sia errato proprio un campo chiave per cui il Record sba-

gliato non è più richiamabile con la procedura di modifica dati.

In tal caso occorre intervenire in comandi diretti (al di fuori della procedura) sul file, oppure se si sta realizzando una procedura per utenti non esperti occorre predisporre un programma che recuperi i dati errati o che li cancelli.

Comandi supplementari

Una volta creato un archivio è possibile modificarne la struttura conservandone, il più possibile e compatibilmente con le modifiche apportate, i dati. È inoltre possibile copiare l'intero file in un altro, in pratica duplicandolo. È possibile copiare la struttura di un file in un altro, che quindi è identico a quello di provenienza come struttura, ma è completamente vuoto di record.

Infine c'è un comando di copia estremamente potente che duplica solo i campi voluti e i record voluti da un archivio origine ad un archivio destinazione. La sintassi di tali comandi è:

COPY FILE <nome del file A> TO <nome del file D>

provoca la duplicazione del file intero, vale per qualsiasi tipo di file e corrisponde all'identico comando DOS, solo che in caso di esistenza del file D, chiede conferma per l'operazione.

COPY STRUCTURE TO <nome del file> produce un nuovo file vuoto con struttura identica a quello di Origine.

RENAME <nome del file> TO <nome del file>

anche questo comando ha un comando DOS corrispondente.

Il comando più potente è il COPY, che al contrario del COPY FILE, agisce dall'interno del file e quindi vale solo per quelli di tipo *.DBF. La sua sintassi è

COPY TO <nome del file> <intervallo> .
FIELDS. <elenco dei campi>
FOR/WHILE. <condizioni>
SDF././DELIMITED.WITH.BLANK/
<delimitazioni> .

Questo è un comando molto potente in quanto permette, con un'unica istruzione sia di duplicare un archivio, che di farne una versione ridotta, con un minore numero di campi, oppure di farne una versione ridotta prelevando solo alcuni dei record presenti, a seconda di certe condizioni.

Per quanto riguarda l'ultima riga di opzioni sono quelle relative al formato. Ovvero è possibile fare delle copie con delimitazioni speciali che ne permettono la lettura da ambienti differenti dal DB, come da BASIC, da DOS, da LOTUS123, da SYMPHONY. Vedremo queste possibilità in una delle prossime puntate.

Con estrema cautela va usato il comando MODIFY STRUCTURE, che in DB2 provoca (ma prima il DB2 avverte) la perdita dei dati. In DB3 non esiste tale avvertimento il che farebbe pensare che sia possibile eseguire tale comando su un file pieno, senza perderne il contenuto. In realtà ho avuto personalmente problemi e ho saputo altrettanto di altri utilizzatori nell'uso dei MODI STRU. Per cui, anche in DB3, conviene seguire il metodo DB2, ovvero:

```
USE <archivio origine>
COPY STRU TO <archivio destinazione>
USE <archivio destinazione>
APPEND FROM <archivio origine>
CLOSE DATABASES
```

In pratica la modifica della struttura si esegue su un archivio vuoto, che, una volta eseguita la modifica, si riempie prelevando i campi che non sono cambiati, dalla vecchia struttura. Se poi si vuole conservare il vecchio nome del file basta cancellare il vecchio file e rinominare il nuovo col nome del vecchio. Cioè:

```
DELETE FILE <archivio origine>
RENAME <archivio destinazione> TO
<archivio origine>
```

Gestione di un archivio in comandi diretti

I comandi da utilizzare per gestire un archivio in maniera diretta sono USE <nome del file>, APPEND, EDIT <numero del record>, DELETE e PACK. Per muoversi all'interno

del file si usano GO <numero del record>, SKIP -/+ <numero di record>.

Per vedere il contenuto di record si può usare il DISPLAY che visualizza il record su cui è posizionato il puntatore, oppure un certo numero di record. Il comando LIST che opera su tutto il file e può operare selezioni per condizioni.

Esiste il comando BROWSE che permette di operare sul file con modalità FULL SCREEN. In pratica appaiono più record contemporaneamente (uno per riga) e si possono modificare direttamente (vedi fig. 6).

Figura 6 - Comando BROWSE. Permette di lavorare in modalità «full screen» sull'archivio. Ovvero appaiono più record contemporaneamente sul video e con i tasti «freccia» ci si può muovere sia in verticale da un record all'altro sia in orizzontale da un campo all'altro. È possibile in tale modalità modificare il contenuto del file.

Record n.	11	ARCDB3				
NO	ME	COGNOME	SEX	DATANASC	PROVNASC	STIPENDIO
LUIGI	ROSSI		M	09/06/58	RM	1200000
MARCO	VERDI		M	20/03/64	PA	1250000
LUDOVICO	BIANCHI		M	29/05/51	LI	2190000
CARLOTTA	GIALLI		F	21/05/45	LI	1875000
MARGHERITA	VIOLA		F	03/10/54	RM	1090000
WALTER	NERI		M	11/01/55	RM	1100000
BENIAMINO	MARRONI		M	29/06/57	RM	1295000
GIUSEPPE	ARANCIO		M	04/09/58	LI	1450000
MARIO	BRUNO		M	20/12/62	TO	1345000
CHIARA	ROSA		F	25/03/60	MI	1456000
ALDO	AZZURRI		M	15/02/56	MI	2009000

In realtà il BROWSE è un po' difficile da maneggiare in quanto dispone di un suo EDITOR, che permette movimenti verticali sullo schermo, che corrispondono a movimenti sul file, e movimenti orizzontali, che corrispondono a movimenti sui campi dello

stesso record. Esistono comandi di inserimento e cancellazione righe (e quindi record).

Le difficoltà crescono e arrivano a rendere il comando BROWSE immangiabile quando la lunghezza del tracciato è tale che il singolo record supera una videata e allora muoversi tra campi, record e schermi diventa problematico.

Gestione di un archivio sotto programma

Un uso tramite comandi diretti di DB2 e DB3 può essere fatto solo da

chi conosce bene tutti i comandi, e solo quando i dati non siano molti e quando non sia indispensabile la correttezza dei dati.

In realtà DB2 e DB3 sono strumenti professionali che possono gestire archivi importanti, sia in termini di dimensioni che in termini di affidabilità dei dati. Vanno quindi realizzati dei programmi per la gestione degli archivi e per la loro utilizzazione.

In particolare diventa indispensabile ricorrere alla programmazione quando occorre mettere in relazione vari archivi tra di loro, ad esempio perché un elemento di un archivio dipende da informazioni che stanno in un altro archivio, oppure quando un campo debba essere sottoposto a controlli particolari.

In questa puntata pubblichiamo un programma che serve per l'immissione di nuovi dati nello stesso file che abbiamo utilizzato in alcuni esempi precedenti. I concetti nuovi sono numerosi e quindi abbiamo preferito inserire l'esempio in un riquadro che illustri i vari comandi. In figura 7 potete vedere un esempio di output del programma.

Nella struttura del file che vogliamo alimentare esistono due campi che vanno controllati, il campo SEX, che può assumere solo i due valori F e M, e il campo PROVNASC (provincia di nascita) che deve corrispondere ad una delle sigle automobilistiche delle province italiane.

L'esempio si può generalizzare, in pratica un campo dell'archivio princi-

```

. USE ARCDB3
. LIST
Record NO    NOME          COGNOME          SEX  DATANASC  PROVNASC  STIPENDIO
  1  LUIGI          ROSSI            M    09/06/58  RM        1200000
  2  MARCO          VERDI            M    20/03/64  PA        1250000
  3  LUDOVICO       BIANCHI          M    29/05/51  LI        2190000
  4  CARLOTTA      GIALLI           F    21/05/45  LI        1875000
  5  MARGHERITA    VIOLA            F    03/10/54  RM        1090000
  6  WALTER         NERI             M    11/01/55  RM        1100000
  7  BENIAMINO     MARRONI          M    29/06/57  RM        1295000
  8  GIUSEPPE      ARANCIO          M    04/09/58  LI        1450000
  9  MARIO         BRUNO            M    20/12/62  TO        1345000
 10  CHIARA        ROSA             F    25/03/60  MI        1456000
 11  ALDO          AZZURRI          M    15/02/56  MI        2009000

. GO 3
. DISPLAY
Record NO    NOME          COGNOME          SEX  DATANASC  PROVNASC  STIPENDIO
  3  LUDOVICO       BIANCHI          M    29/05/51  LI        2190000

. SKIP 2
Record n.    5
. ? RECNO()
5
. SET HEADING OFF
. DISP
5  MARGHERITA  VIOLA            F    03/10/54  RM        1090000

. GO BOTTOM
. DISPLAY
11  ALDO          AZZURRI          M    15/02/56  MI        2009000

. GO TOP
. ? RECNO()
1
. DISPLAY
1  LUIGI          ROSSI            M    09/06/58  RM        1200000

. SKIP -1
Record n.    1
    
```

Figura 5 - Ginnastica all'interno dell'Archivio. Quando si apre un archivio con il comando USE <nome archivio> il puntatore si posiziona sul primo record. Per cui se si imposta il comando DISPLAY viene visualizzato il primo RECORD. Esistono vari comandi per muoversi su e giù nel file.

pale rimanda ad un archivio secondario che contiene i dati di decodifica del campo. Si introduce il concetto di lavoro su più archivi aperti contemporaneamente e del passaggio da uno all'altro, tramite un semplice comando. Il passaggio comporta il fatto che pur

essendo aperti più file, in un dato momento si può lavorare solo su un file, e in questo momento sono riconosciuti dal comando REPLACE solo i numeri dei campi del file in uso.

Il problema più grosso quando l'applicazione richiede numerosi file aper-

ti è proprio costituito dal sapere in ogni momento quali sono i file aperti e quali di questi sia in uso. In fase di debug dei programmi risulta particolarmente utile il comando DISPLAY STATUS che fa la «fotografia» in quell'istante della situazione file. **MC**

I primi programmi in dBASE

Iniziamo con questa puntata a mostrare dei listati di programmi in dBASE. Il primo della serie è un programma che gestisce l'immissione di nuovi record in un archivio. Il listato è abbastanza elementare in quanto non abbiamo inserito sofisticazioni di programmazione, ma poiché lavora su due archivi è già abbastanza avanzato per il neofita o per chi legga l'articolo senza provare dopo sulla macchina gli esempi. Fortunatamente il DB non ha istruzioni di salto, il che facilita la comprensione del listato di un programma che si svolge sempre in sequenza.

All'inizio del programma si eseguono i vari settaggi permessi dalle istruzioni SET, qui ad esempio si disabilita la ripetizione dei comandi di assegnazione e si abilita l'uso del carattere «due punti» come separatore di campi.

Si aprono due archivi, il primo è quello usato negli esempi precedenti, il secondo è un file la cui struttura è formata solo da due campi:

SIGLA
CITTA

In tale archivio in pratica sono contenute tutte le sigle automobilistiche e a ciascuna di queste è associato il nome della città. Tale archivio ha un indice, vedremo nel prossimo numero di che si tratta, per ora diciamo solo che abbiamo creato una chiave di ricerca nella tabella costituita dalla sigla automobilistica, per cui data una sigla il DB provvede a posizionarsi sul record corrispondente.

L'istruzione

DO WHILE <condizione> ENDDO

genera un loop chiuso dall'ENDDO, dal quale si esce solo quando non si verifica più la condizione iniziale o quando c'è una istruzione di uscita tipo RETURN. In pratica vogliamo continuare ad immettere dati fin quando non diamo un comando di fine.

CLEAR pulisce lo schermo (in DB2 è

```
* 01/12/1985 CONTROLLO IMMISSIONE ( db3 )
SET TALK OFF
SET DELIMITER ON
CLEAR
* archivio principale
SELECT 1
USE ARCDDB
* tabella esterna
SELECT 2
USE PRVDB3 INDEX PRVIND
SELECT 1
*
DO WHILE .T.
CLEAR
VAR1=""
@ 1,1 SAY "NOME"           " GET VARI1 PICTURE "!!!!!!!!!!!!"
READ
* controllo fine input
IF VAR1=""
RETURN
ENDIF
APPEND BLANK
REPLACE NAME WITH VAR1
@ 2,1 SAY "COGNOME"       " GET COGNOME PICTURE "!!!!!!!!!!!!"
READ
* controllo dato
DO WHILE .NOT. SEX*"MF"
@ 3,1 SAY "SESSO"         " GET SEX PICTURE "!"
```

ERASE) e APPEND BLANK predispone un nuovo record, per ora vuoto.

La sintassi tipica del comando di immissione è

@ X,Y SAY <messaggio> GET <nome campo> PICTURE <formato> READ

dove

@ X, Y sono i valori numerici delle coordinate dello schermo sul quale ci si posiziona

NOME	:CLAUDIO	:
COGNOME	:ROSSINI	:
SESSO	:M:	:
DATA DI NASCITA	:11/12/37:	:
PROVINCIA	:TO: TORINO	:
IMPORTO	: 1,200,000:	:

Figura 7 - Esempio di output.

SAY permette di scrivere un messaggio sul video, e tale messaggio, come nella migliore tradizione, va posto tra virgolette;

GET permette l'input o su un campo o su una variabile che però va preventivamente definita.

PICTURE permette una notevole varietà di formati di immissione, ad es. il punto esclamativo esegue la conversione dei caratteri immessi in caratteri maiuscoli.

Il GET <nome del campo> viene attivato dalla istruzione di READ seguente. Ogni READ può riguardare più GET, ma se del campo in input si vuole eseguire all'istante un controllo, occorre dedicargli una READ.

Il campo NOME non viene caricato direttamente da una specifica coppia di GET & READ, ma si appoggia ad una variabile di comodo VAR 1, questo perché vogliamo controllare anche la fine immissione dando un invio a vuoto sul campo nome. In pratica si definisce la variabile VAR 1 di lunghezza pari al campo nome, si fa la GET della variabile, si con-

trolla se viene dato un invio a vuoto. Se è stato dato un invio a vuoto si esegue il comando RETURN, altrimenti si «rimpiazza» il campo nome con il contenuto della VARI.

Nessun problema per il COGNOME, in quanto è senza controlli. Invece per controllare che il SESSO sia o F o M, viene realizzato un loop (con DO WHILE.. ENDDO) dal quale si esce solo se il campo SEX contiene o S o F. Questo lo fa una comodissima funzione logica (di ricerca di sottostringa) che indica se in una certa variabile esistono o meno certi caratteri.

Si crea un loop anche per il controllo della provincia, ma generalizziamo il concetto di controllo attivando un Flag che fa iniziare un loop dal quale si esce solo se il controllo è andato a buon fine e quindi si è azzerato il Flag.

Nel nostro caso il controllo consiste nella ricerca della sigla della provincia nell'archivio secondario. Se la ricerca provoca la condizione logica di fine del file vuol dire che non si è trovata la sigla e quindi si invia un messaggio di non trovato. Se la sigla è presente nell'archivio secondario viene scodificata scrivendo la città e viene disattivato il Flag, per cui si può uscire dal loop.

Usciti dal loop la prima cosa da fare è di riposizionarsi sul file primario e di rimpiazzare il campo PROVNASC con la variabile di comodo usata.

L'immissione dell'importo ci permette di vedere un altro tipo di PICTURE per dati numerici. Il programma si chiude con un loop di attesa del tutto uguale ai FOR..NEXT del Basic e con l'ENDDO che chiude il loop principale iniziato con il comando DO WHILE.T.

Questo DO WHILE.T. ... ENDDO è un ciclo senza condizione iniziale e dal quale si esce solo con un comando di uscita dal programma che è il RETURN.

```
READ
ENDDO
@ 4,1 SAY "DATA DI NASCITA " GET DATANASC
READ
* controllo in una tabella esterna
FL=1
DO WHILE FL=1
VAR2=""
@ 5,1 SAY "PROVINCIA"     " GET VAR2 PICTURE "!"
READ
SELECT 2
FIND "VAR2"
IF EOF()
@ 5,25 SAY "NON TROVATA"
ELSE
@ 5,25 SAY CITTA
FL=0
ENDIF
ENDDO
SELECT 1
REPLACE PROVNASC WITH VAR2
@ 6,1 SAY "IMPORTO"       " GET STIPENDIO PICTURE "###,###,###"
READ
CN=1
DO WHILE CN<100
CN=CN+1
ENDDO
ENDDO
```


SCALDA IL JOYSTICK E GASATI CON



...Compilation... ..

JACKSON SOFT

ogni mese
IN EDICOLA



**GRUPPO
EDITORIALE
JACKSON**

Milano-Londra-Madrid-San Francisco

IL VERO GIOCO COMINCIA ADESSO

1971-1986

EDP USA COMPIE 15 ANNI

EDP USA 86



MILANO 4-7 FEBBRAIO 1986

MOSTRA E SEMINARI

EDP USA compie 15 anni. Quindici anni di mostre che hanno costituito un "ponte" fra la tecnologia informatica americana e la creatività applicativa italiana.

EDP USA 86 rappresenta il momento di incontro di queste forze al più alto livello specialistico.

I 6.000 mq. di area espositiva ospiteranno, nei quattro giorni della manifestazione i produttori americani più rappresentativi di tutti i comparti dell'informatica, una sezione Computer Graphics e una Trade Mission ufficiale di dieci membri specializzati in software proveniente dagli Stati Uniti. Nel salone dei congressi si svolgeranno

tre seminari presieduti da eminenti personalità dell'informatica:

5 febbraio: "Infocenter: Strumenti e Organizzazione, una Realtà per il Futuro".
Coordinatore: Prof. Giancarlo Martella, docente di Trattamento dell'Informazione nell'Impresa presso il Politecnico di Milano.
Quota di partecipazione: Lit. 75.000

6 febbraio: "Un viaggio dall'Invenzione del Transistor alla Intelligenza Artificiale".
Coordinatore: Prof. Marco Somalvico, ordinario di Complementi di Programmazione e di Robotica Industriale dell'Istituto di Elettronica del Politecnico di Milano.

Quota di partecipazione: Lit. 150.000, colazione inclusa.

7 febbraio: "Personal Computer e CAD: nuove frontiere da esplorare".
Coordinatore: Ing. Roberto Favero, consulente di informatica.
Quota di partecipazione: Lit. 75.000.

Le quote di partecipazione dovranno pervenire a mezzo assegno intestato a USIMC entro il 27 gennaio. L'ingresso alla mostra è riservato agli operatori del settore.

Gli studenti sono ammessi nel pomeriggio dei giorni 4 e 7 febbraio previa prenotazione e pagamento di un biglietto ingresso di Lit. 3.000.
Orario di apertura della mostra: 9.00 - 18.00.



CENTRO COMMERCIALE AMERICANO

VIA GATTAMELATA, 5 - 20149 MILANO - TELEPHONE 46 96 451/2 - TELEX 330208 USIMC-I