

dBASE IV e Paradox 3.0

Pratica con il linguaggio QbE

di Francesco Petroni

La maggior parte delle novità che presenta il dBASE IV nei confronti del suo predecessore dBASE III riguardano le modalità di lavoro assistito, che, a detta più dei concorrenti, che degli utilizzatori del prodotto, erano un po' carenti.

La tendenza che si riscontra non solo nei prodotti, ma anche nei sistemi operativi (vedi DOS 4.0 oppure OS/2 Presentation Manager) è quella di offrire interfacce «amichevoli» in cui anche l'utente inesperto sia in grado di lavorare in quanto tutto quello che può fare gli viene via via suggerito dall'ambiente assistito

In questa ottica il classico Dot Prompt (puntino lampeggiante del dBASE), che indica «attesa di un comando» diventa uno strumento troppo antiquato.

La risposta dei tecnici della Ashton Tate è stata quella di migliorare sia i comandi fondamentali, e quindi richiamabili anche dal puntino, sia l'ambiente assistito, che con dBASE IV offre nume-

rose strutture specializzate a seconda della singola attività da svolgere.

Le novità fondamentali sono il Control Center, centro di smistamento delle varie attività assistite, e, all'interno di questo, la opzione QUERY, che introduce, anche nel dBASE IV, un linguaggio di interrogazione degli archivi che ha avuto un ottimo successo di critica e un buon successo di pubblico nel Paradox, della Borland, e che si chiama Query by Example.

Vanno subito precisate due cose.

Il modo di lavorare «nativo» del dBASE IV è comunque costituito dai comandi richiamabili dal Dot Prompt e quindi tutte le strutture assistite, e quindi anche il Query by Example, producono uno o più comandi comunque digitabili dal puntino.

Il QbE nel Paradox è l'unico linguaggio di interrogazione degli archivi (che in Paradox si chiamano TABLES) ed è quindi efficientissimo. Il QbE del dBASE IV è invece un confezionatore di una sequenza di comandi puntino (che tralaltro vengono memorizzati in un file di desinenza .QbE).

Non essendo quindi un'alternativa, ma solo una sovrastruttura, rispetto al Dot Prompt, non è particolarmente brillante in termini di prestazioni, né particolarmente ricco in termini di elementi sintattici. Ciò nonostante può trovare un range di utilizzabilità. Di questo parleremo comunque alla fine dell'articolo.

La filosofia del QbE

Il Query by Example si basa sul concetto che l'utente deve impostare, in una schermata opportunamente predisposta, un «esempio» di quello che vuole.

L'operazione si divide in due fasi, la prima, in cui l'utente imposta, secondo la sintassi QbE, l'interrogazione, e la seconda, in cui il prodotto esegue la Query e fornisce la risposta.

CLIENTI					
Record	CCOD	CMON	CTIP	CSCO	CIMP
1	C003	SMITH	C	0.90	0
2	C002	JOHNSON	B	0.85	0
3	C001	PERKINS	A	0.80	0
4	C004	WILLIAMS	A	0.90	0
5	C005	BLACK	D	0.80	0
6	C006	WHITE	D	0.85	0
7	C007	MASON	A	0.80	0
8	C008	WELLS	D	0.85	0
9	C009	FORD	D	0.75	0

ARTICOLI					
Record	ACOD	ATIP	ADES	APRE	AIMP
1	A004	X	MATITA	500	0
2	A002	Y	PENNA	1000	0
3	A003	Z	QUADERNO	1500	0
4	A001	Y	GOMMA	300	0
5	A007	Y	QUINTERNO	300	0
6	A008	X	TEMPERINO	500	0
7	A009	Z	COLLA	1200	0
8	A005	Y	CANCELLINO	700	0
9	A006	X	PUNTINE	500	0

ORDINI						
Record	ONUM	OART	ODAT	OCLI	OQUA	OIMP
1	001	A006	01/01/89	C001	5	
2	011	A004	01/02/89	C002	12	
3	012	A005	01/03/89	C004	10	
4	003	A003	01/02/89	C002	30	
5	002	A004	01/03/89	C002	10	
6	004	A002	01/02/89	C004	5	
7	005	A001	01/01/89	C003	12	
8	006	A009	01/03/89	C002	10	
9	007	A001	01/04/89	C002	6	
10	008	A001	01/02/89	C008	10	
11	009	A005	01/01/89	C001	20	
12	010	A001	01/03/89	C004	15	
13	013	A004	01/04/89	C005	20	
14	014	A008	01/04/89	C007	10	
15	017	A007	01/03/89	C009	30	
16	015	A005	01/04/89	C006	12	
17	019	A005	01/03/89	C004	8	
18	018	A007	01/03/89	C008	18	
19	020	A002	01/02/89	C007	8	
20	016	A009	01/02/89	C004	12	

Figura 1 - QbE - Gli Archivi di prova. Si tratta di tre archivi, ARTICOLI, CLIENTI e ORDINI. Ordini è collegato, tramite il codice Articolo e il codice Cliente ai primi due. Con questi tre archivi è possibile testare abbastanza a fondo, anche se non completamente, le problematiche connesse all'uso del QbE

Nel dBASE IV, come detto, la risposta in pratica comporta la composizione di una serie di comandi Dot Prompt e la loro esecuzione in sequenza, nel Paradox, invece, viene eseguita una procedura, che si basa su un processo di tipo euristico, e che permette una ottimizzazione delle prestazioni.

Il QbE lavora in una videata in cui compaiono le varie strutture (Skeleton) degli archivi aperti. Le strutture vengono visualizzate «in orizzontale» con un grafismo che presenta sopra il nome del campo e sotto di questo uno spazio in cui l'utente digita l'Example. In questo spazio libero si possono mettere alcuni elementi sintattici raggruppabili in più categorie.

- Condizioni di selezione.
- Codifiche per creare i vari tipi di ordinamento.
- Codifiche per creare in uscita campi calcolati.
- Comandi individuali di calcolo.
- Comandi insiemistici di calcolo.
- Nome da dare al campo in uscita.
- Example per mettere in relazione i vari archivi.
- Operatori che lavorano sulle relazioni.

Il risultato di una Query può essere anche una operazione di aggiornamento degli archivi sia in termini di creazione di nuovi record che di modifica di record già presenti che, infine, in termini di cancellazione di record. Esistono quindi specifiche codifiche.

La differenza esteriore fondamentale (oltre a quella meno evidente, ma più importante, riguardante il modo con cui il prodotto esegue «internamente» la Query) tra i due QbE è che il dBASE IV utilizza due strutture in più rispetto a quelle rappresentanti gli archivi, quella dei campi calcolati e quella delle condizioni, mentre nel Paradox si utilizzano solo le strutture dell'archivio, in cui, ad esempio, si possono impostare anche le codifiche necessarie per creare un campo calcolato. Inoltre nella videata Query del dBASE IV appare anche lo Skeleton dell'archivio in uscita (sta in basso e si chiama VIEW), per cui le varie impostazioni che lo riguardano (ad esempio il nome da far assumere ai campi) vi possono essere impostate direttamente e più semplicemente.

Il nostro caso studio

Usiamo tre archivi, uno di Articoli, uno di Clienti e uno di Ordini, che fa riferimento sia agli Articoli che ai Clienti, di cui vediamo struttura e contenuto in figura 1.

È un caso semplificato al massimo (ad esempio ogni ordine ha un solo articolo) ma che ci serve per mostrare un buon numero di esempi di Query. Le illustrazioni sono riferite al singolo prodotto con il quale sono state realizzate, ma la nostra intenzione è di fare comunque un discorso generico sul QbE, valido indipendentemente dal «dialetto» utilizzato.

Il Query by Example è infatti «filosoficamente» uguale nei due prodotti, quindi in certi casi è possibile eseguire degli esempi ambivalenti, ma differisce nella sintassi (tanto per fare un esempio l'operatore EVERY del dBASE in Paradox c'è ma usa il simbolismo «!»). Tornando al caso studio i collegamenti tra i tre archivi, che comunque analizzeremo a fondo tra un po' quando li imposteremo nella Query, sono evidenti. È anche evidente che una volta impostate correttamente le relazioni sarà possibile manipolare dati provenienti dai tre archivi. Ad esempio sarà possibile eseguire il calcolo dell'importo dell'ordine il cui risultato è dato dalla Quantità venduta per il Prezzo del

l'Articolo meno lo Sconto del Cliente (lo sconto è per semplicità di calcoli espresso come complemento a 1, quindi 90 significa sconto del 10 per cento).

Questi tre fattori provengono ciascuno da uno dei tre archivi.

Una volta definite correttamente le relazioni è il prodotto che garantisce l'allineamento tra i tre archivi e quindi sarà possibile eseguire qualsiasi manipolazione. Buona parte di queste saranno eseguibili in modalità interattiva (e quindi non scrivendo alcun programma) dall'ambiente QbE. Ne forniamo un elenco di esempio:

- archivio ordini completato con i dati di articoli e clienti;
- ordinato per cliente e per data, o su qualsiasi altro campo presente in uno qualsiasi dei tre articoli;
- somma importi per cliente ordinati per cliente;
- somma importi per cliente ordinati per somma importi;
- somma importi per articolo ordinati per articolo;

Figura 2 - Paradox 3.0 Sintassi del Query. Se su un campo dello Skeleton di un archivio in lettura si preme F6, apparirà un segnetto, e il campo sarà riportato nella vista. Con Alt F6 si imposta un ordinamento discendente. La stringa «AS NO-ME» serve per cambiare in uscita il nome del campo. La stringa A, posta sotto un campo, ha il significato di condizione di selezione. L'operatività sulla Query è semplificata quanto la sua simbologia.

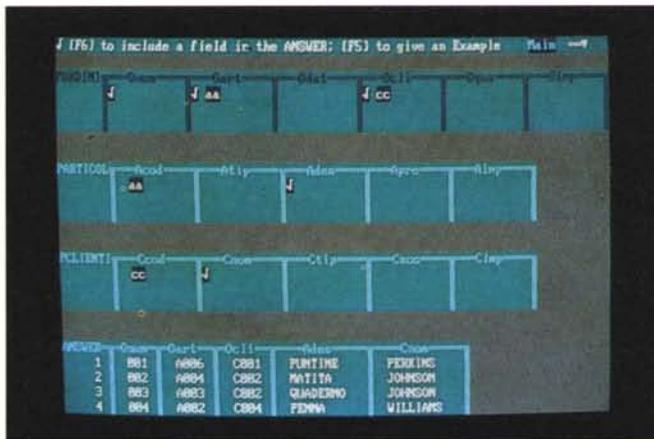
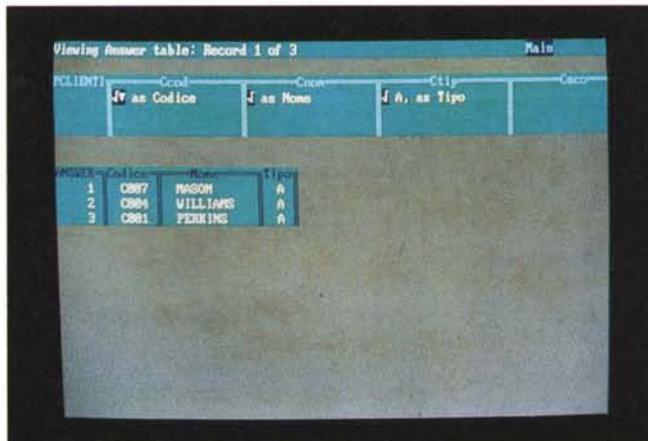


Figura 3 - Paradox 3.0 Link tra gli archivi. I collegamenti tra gli archivi si ottengono utilizzando la tecnica degli Example. In pratica si scrive, al di sotto dei campi che realizzano il collegamento, una stessa stringa, facendola precedere dal tasto F5 (altrimenti si confonderebbe con la selezione). Una volta impostato il collegamento è possibile selezionare campi dai vari archivi collegati.

- somma importi per articolo ordinati per somma importi;
- sottototali per qualsiasi campo su cui fare un raggruppamento;
- ecc.

Gli esempi

Descriveremo ora, suddividendole in

grosse famiglie, le operazioni possibili con il Query cominciando anche a riferirci alle illustrazioni a corredo dell' articolo.

01 - Collegamento tra gli archivi.

I campi in comune sui quali appoggiare il collegamento, sono il codice Clien-

te, che si chiama OCLI nell'archivio Ordini e CCOD nell'archivio Clienti, e il codice Articolo, che si chiama OART nell'archivio Ordini e ACOD nell'archivio ARTICOLI.

Occorre mettere delle variabili «Example» uguali nelle due coppie di campi e sarà il Query che provvederà ad allineare i due archivi secondari in funzione dei codici Clienti ed articoli trovati nei primari (figg. 2, 3, e 9).

Il dBASE IV permette anche il collegamento via opzione di menu che in pratica inserisce proprie variabili di Example (LINK1, LINK2, ecc.) ma che equivalgono a qualsiasi altro Example.

Una volta attivati i collegamenti è possibile avere in uscita campi presi dai tre archivi.

Nel nostro esempio si attiva una relazione tra Ordini e Clienti e quindi in ogni Ordine c'è un codice Cliente che (presumibilmente) è presente anche nell'archivio Clienti.

In caso di assenza (ovvero il codice Cliente scritto nell'Ordine non trova corrispondenza nell'archivio Clienti) non si verifica una situazione di errore. I campi alfanumerici prelevati dall'archivio Clienti risultano vuoti, mentre quelli numerici danno valore zero (questo per non danneggiare eventuali espressioni di calcolo).

Il Query gestisce anche l'eventualità che si vogliono visualizzare anche i dati dell'archivio secondario che però non sono presenti in nessuno degli ordini. Ad esempio elencare anche i Clienti che non hanno fatto Ordini o Articoli che non siano stati ordinati. Questo si fa con l'espressione EVERY in dBASE e con il simbolo «!» in Paradox.

02 - Collegamento 1 a molti

Nell'esempio ora visto l'archivio principale è quello degli Ordini. Questo è in relazione con gli altri due in quanto ad ogni codice Articolo presente nell'Ordine corrisponde un Articolo nell'archivio Articoli ed analogamente verso l'archivio Clienti.

All'articolo dell'archivio Ordini corrisponde un solo articolo nell'archivio Articoli, ma viceversa ogni Articolo di quest'ultimo è presente in molti ordini. Questa situazione si chiama relazione Molti a 1, ed è diffusissima.

Se ora invertiamo l'ordine degli archivi, aprendo come principale quello degli Articoli e come secondario quello degli Ordini ribaltiamo la situazione nel senso che ad ogni Articolo presente nell'anagrafico corrispondono nessuno, uno o più Ordini. Questa situazione si chiama, come ovvio, 1 a Molti.

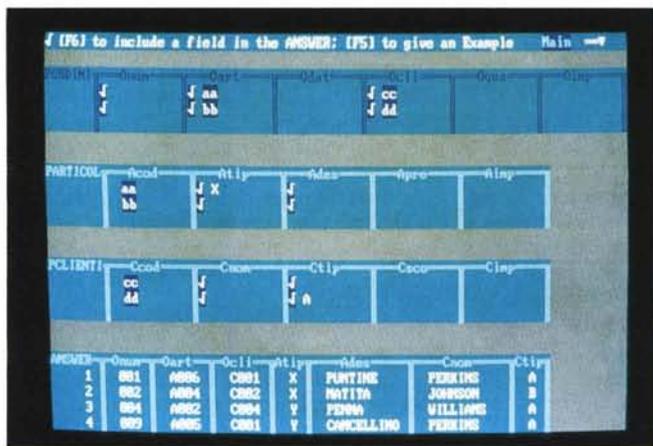


Figura 4 - Paradox 3.0 Query complessa. Una volta collegati i vari archivi, con gli Example, continua ad essere possibile impostare selezioni, che possono lavorare indifferentemente sui vari scheletri. Nella foto vediamo due condizioni in OR, e quindi, come spiegato nel testo, scritte su due righe. Il tipo dell'articolo deve essere X «OPPURE» il tipo del cliente deve essere A.

Figura 5 - Paradox 3.0 Campo calcolato. Esempio di utilizzo dell'Example per realizzare in uscita un campo calcolato. Sotto il campo Apre dell'archivio Articoli è stato impostato un campo calcolato dato dal prodotto del campo Oqua (quantità venduta presente nell'archivio Ordini) per il campo Apre (prezzo dell'articolo). Anche scrivendo la formula di calcolo occorre premere F5 prima di impostare le stringhe scelte come Example.

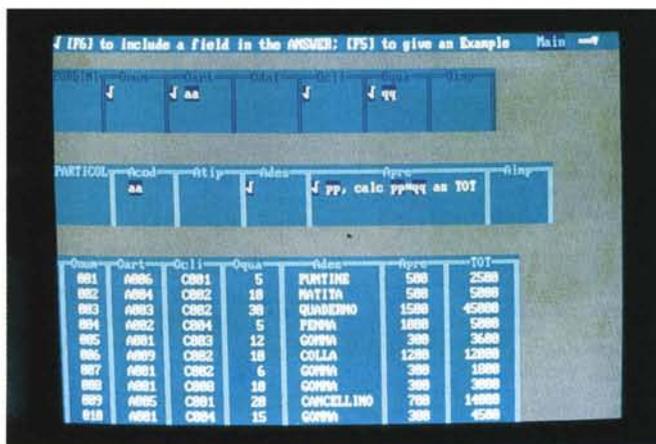
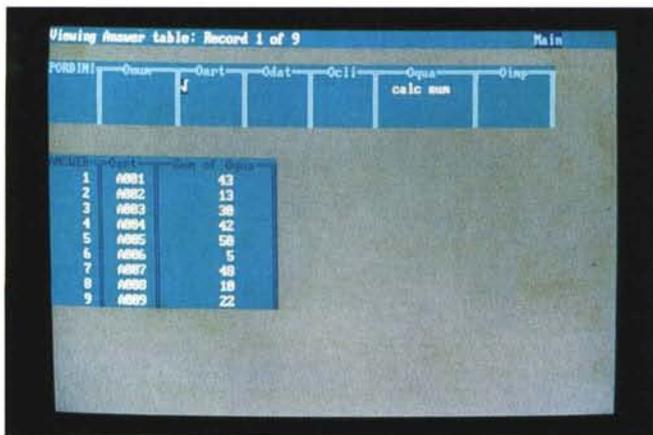


Figura 6 - Paradox 3.0 Comandi di raggruppamento. Altro esempio della sinteticità del Query by Example. In questo caso vogliamo totalizzare le quantità vendute per articolo. Basta mettere il checkmark (con F6) sotto il campo Oart e l'espressione CALC SUM sotto il campo sul quale eseguire i sottototali. Di esempi di analogia semplicità se ne potrebbero fare centinaia.



Se si imposta la Query, collegando tramite gli Example, i due codici Articoli, si può ulteriormente specificare il comportamento della Query rispetto a questa situazione.

Sia il dBASE che il Paradox tengono conto del tipo di relazione nel senso che per ogni codice articolo visualizzano TUTTI gli Ordini in cui sono presenti. Si possono però inserire delle espressioni che alterano il risultato.

EVERY (visto prima) permette la visualizzazione degli Ordini in cui ci sia un codice Articolo che non trova corrispondenza nell'anagrafico.

FIRST, limita la visualizzazione al solo PRIMO Ordine che contiene quel codice Articolo.

03 - Selezioni

Il risultato in uscita da una operazione di Query appare a tutti gli effetti come un altro archivio su cui si può intervenire con funzioni di Editing o di Browsing.

Nella sintassi del Query è anche possibile inserire dei parametri di selezione, che una volta collegati gli archivi, possono agire su qualsiasi campo di qualsiasi archivio collegato.

La modalità di selezione è simile a quella utilizzata dalla Zona Condizione del Lotus 123 ed è quindi semplice per cose semplici ma diventa sofisticata per selezioni complesse.

In caso di selezione su un solo campo basta scrivere la condizione sotto il campo corrispondente.

Se la condizione è di semplice eguaglianza si scrive la stringa o il numero (in dBASE la stringa va scritta tra apici). Valgono per le stringhe e per i numeri i vari operatori relazionali (minore, maggiore, ecc.).

Il caso di condizioni su più campi, se queste sono in AND si scrivono sulla stessa riga, se sono in OR su righe successive. Ad esempio:

OART OCLI
0001 C001

significa articolo 0001 E cliente C001

OART OCLI
0001 C001

significa articolo 0001 O cliente C001

OART OCLI
0001 C001
0002 C001

significa articolo 0001 O cliente C001.

Come al solito il limite superiore della complessità della selezione sta più nella nostra mente che non riesce a formalizzare gli Example, che nel prodotto (figg. 4, 11).

04 - Campi calcolati

In tutte le applicazioni di una certa entità esistono i cosiddetti campi calcolati, in pratica delle informazioni che dipendono del tutto dalle altre, al punto che in genere non si inseriscono per nulla nell'archivio in quanto vengono ricalcolate ogni volta che servono. In

questo caso si chiamano campi VIRTUALI.

La Query è la situazione «ideale» nella quale costruire ed utilizzare campi calcolati (figg. 5 e 12).

Nella nostra applicazione un campo calcolato è sicuramente l'importo dell'ordine la cui espressione matematica (nella nostra semplificazione che non

Figura 7 - Paradox 3.0 Help sul campo calcolato. Per migliorare l'operatività è indispensabile disporre di efficaci help in linea che propongano sintetiche spiegazioni sul da farsi ed efficaci esempi su come farlo. Qui vediamo come si costruisce un campo calcolato in un Query del Paradox.

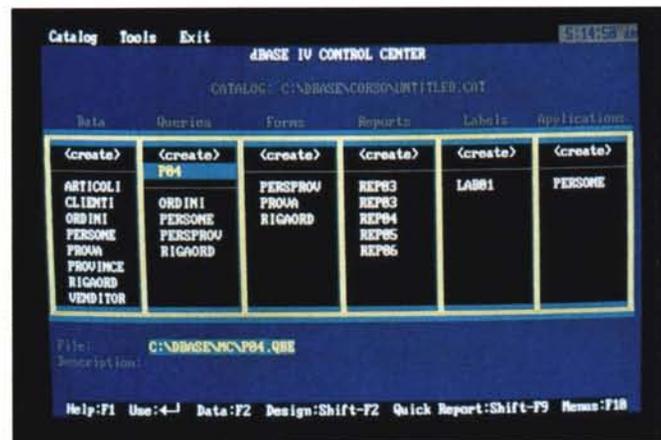
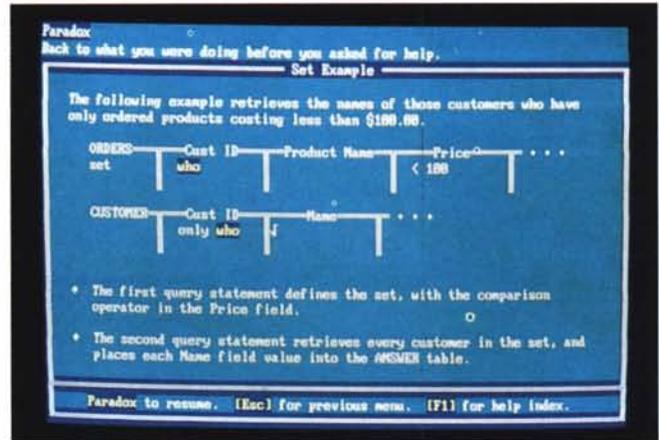


Figura 8 - dBASE IV Il Control Center. Nel dBASE IV l'oggetto Query assume la stessa dignità di un archivio monostruttura tradizionale. Sulla Vista prodotta dall'operazione di Query si possono costruire Maschere oppure eseguire Report. Se ad esempio occorre ottenere un dato come elaborazione di più dati in genere si può scegliere se eseguire il calcolo a livello di Query o direttamente sul Report.

Figura 9 - dBASE IV Skeleton. Questa videata, un po' rimaneggiata, mostra come, nell'ambiente Query del dBASE IV, al contrario di quanto avviene nel Paradox, appare già la struttura della View, sulla quale si possono eseguire già alcune operazioni, come l'imposizione del nome delle zone.



prevede né righe d'ordine né calcolo dell'IVA) è il prodotto tra la Quantità e il Prezzo, decurtato dello Sconto.

Questi tre dati vengono prelevati, per mezzo della Relazione impostata, dai tre archivi in cui risiedono, per cui non c'è nessuna difficoltà concettuale ad impostarne la formula. Le strade tra dBASE e Paradox divergono in questo caso sensibilmente dal punto di vista operativo.

Col dBASE IV il campo calcolato si imposta in uno specifico Skeleton in cui si imposta la formula di calcolo e nella quale si attribuisce il nome da dare al nuovo campo.

Nel Paradox, invece, non esistono altri Skeleton oltre a quelli degli archivi e quindi l'espressione di calcolo va impostata in uno di questi Skeleton, utilizzando lo spazio sotto un campo qualsiasi. Il nome del campo, al pari di qualsiasi altro nome, si imposta con la specifica AS. Ad esempio:

OQUA*APRE*CSCO as IMPORTO

05 - Selezioni avanzate

In caso di calcoli e/o selezioni complesse è conveniente ricorrere agli «Example», attraverso i quali è più semplice scrivere formule o espressioni logiche.

Se, all'interno dello Skeleton, in corrispondenza di un campo, si scrive una variabile (ad es. X), questa è poi utilizzabile in luogo del nome del campo, in quanto assume proprio il significato di nome del campo.

Sulla base di questo principio è chiaro che l'Example diventa uno strumento operativo, all'interno dell'ambiente Query, di estrema importanza, paragonabile all'importanza che assumono le variabili in un linguaggio di programmazione.

Diventa anche difficile specificare i limiti applicativi della variabile Example, che possono essere molto spinti.

06 - Sommatorie

Poiché l'esecuzione della Query provoca uno scorrimento ed una elaborazione degli archivi è naturale che sia possibile inserire degli operatori globali di calcolo (fig. 6).

In pratica, sempre all'interno dello Skeleton ed in corrispondenza del campo interessato vanno scritte delle parole Chiave, che possono essere, al solito, COUNT, SUM, AVG, MIN e MAX, se il campo è di tipo numerico, oppure solo COUNT, MIN e MAX se il campo è alfanumerico o data. Il risultato, nel caso più semplice, è una vista che riporta, in una sola riga, i valori richiesti.

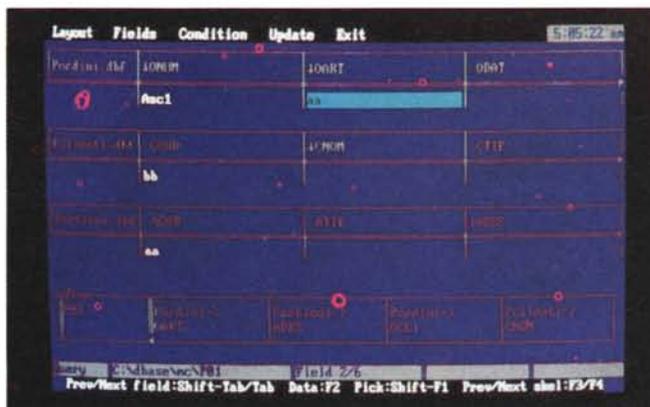
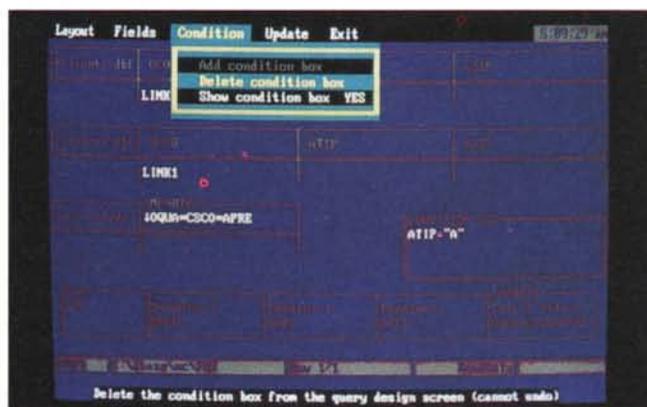


Figura 11 - dBASE IV Calculation Box e Condition Box. Il dBASE, al contrario del Paradox, dispone di due Skeleton operativi in più rispetto a quelli rappresentanti gli archivi. Nel primo si possono impostare i vari campi calcolati e nell'altro condizioni di selezioni complesse che non si riuscisse ad impostare direttamente negli Skeleton.



È altresì evidente che è possibile eseguire raggruppamenti, ma di questo parliamo tra un po', dopo aver esaminato gli strumenti di ordinamento.

07 - Ordinamenti

Altro strumento operativo interno alla Query è quello che permette di mettere nell'ordine voluto l'archivio virtuale in uscita. E sono possibili, sia in dBASE che in Paradox, più livelli di ordinamento.

In dBASE occorre inserire, al solito in corrispondenza del campo dello Skeleton, le parole Asc1, Asc2, Desc1, ecc. il cui significato è intuitivo. L'operazione si può, un po' più facilmente, eseguire da menu.

In Paradox le parole Chiave sono più sintetiche e in ogni caso, quando cioè non si specifica altro, la vista in uscita viene ordinata secondo il valore del primo campo.

08 - Raggruppamenti

Se l'archivio virtuale è ordinato secondo uno o più campi si può procedere ad operazioni di raggruppamento, ad esempio per calcolare totali parziali di gruppo, sottogruppo, ecc.

In dBASE basta scrivere GROUP BY

sotto il campo sul quale si vuol operare il raggruppamento e poi si possono scrivere i comandi di calcolo, ad esempio SUM, sotto i campi dei quali si desidera ottenere i valori di gruppo.

Ovviamente si può estendere il concetto di raggruppamento a più livelli successivi.

L'operatore Group By prevale su quello di ordinamento, ma è evidente che è meglio evitare di metterli in contrasto tra di loro.

È anche possibile utilizzare la specifica UNIQUE che traslascia gli elementi duplicati. Ad es. CNT UNIQUE non conta i campi di contenuto duplicato.

09 - Organizzazione dell'archivio in uscita

Abbiamo in precedenza asserito che il risultato di una Query è un archivio virtuale. In questo archivio i vari campi possono essere nominati ed organizzati in maniera del tutto autonoma rispetto agli archivi dai quali derivano.

In dBASE si lavora direttamente sullo Skeleton in uscita, ed è su questo che si scrivono i nuovi nomi dei campi e si opera con funzioni di spostamento, cancellazione e inserimento dei campi.

Il Paradox invece non usa Skeleton in uscita e quindi i nomi dei campi si

Figura 10 - dBASE IV Ambiente Query. La massima occupazione possibile è costituita da tre Skeleton di tre archivi in lettura e dallo Skeleton della View. Poiché poi la larghezza di ciascun campo non è modificabile per ciascuno dei tre Skeleton appaiono contemporaneamente solo tre campi. Il lavoro con più strutture e con archivi molto complessi diventa pesante.

Figura 12 - dBASE IV
Esempio di condizione
AND e OR.

Chi conosce bene il Lotus 123 ritrova in questa modalità operativa, con la quale si impostano più condizioni su campi dei vari archivi tra loro collegati, la stessa tecnica della Zona Criteri dell'123, in cui condizioni sulla stessa riga stanno in AND e condizioni su più righe in OR.

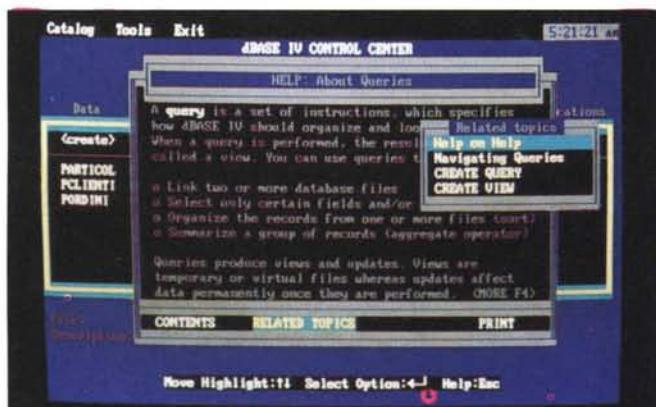
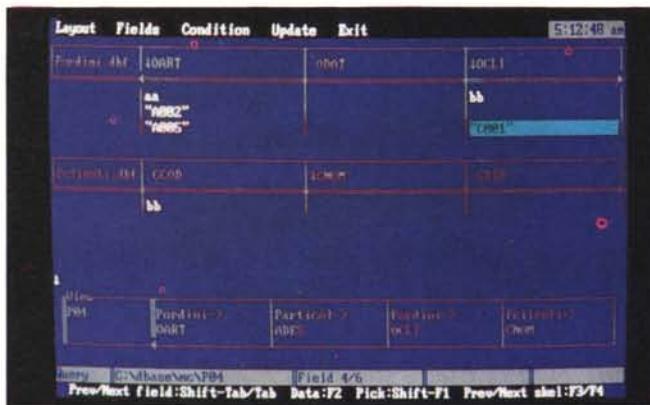


Figura 13 - dBASE IV
Help in linea.

Nel dBASE IV l'Help in linea occupa una finestra sul video, ed è quindi, almeno dimensionalmente, più limitato di quello del Paradox. Inoltre dovendo trattare più argomenti, in quanto il dBASE è sicuramente più complesso e più ricco di funzionalità, anche alternative tra di loro, risulta su un singolo argomento meno approfondito ed esauriente.

danno direttamente sugli Skeleton di partenza, mentre le operazioni di «maquillage» dell'archivio in uscita si eseguono sulla Table prodotta dall'esecuzione della Query.

10 - Relazioni su se stesso (Self Join) e altri usi avanzati

Il linguaggio Query possiede, specie nella versione Paradox, numerosi altri comandi che permettono di risolvere facilmente altre problematiche classiche nella complessa materia che si chiama gestione di un Data Base.

Data la mancanza di spazio siamo costretti per ora a sorvolare, anche perché molti di questi comandi particolari richiederebbero la predisposizione di archivi d'esempio costruiti ad hoc e che occorrerebbe anche descrivere.

Ad esempio sia dBASE che Paradox permettono di relazionare un archivio con se stesso (nel dBASE III era impossibile aprire lo stesso archivio in due differenti aree di lavoro), ad esempio un archivio Personale in cui per ogni dipendente c'è anche un campo in cui inserire la matricola del superiore, che a sua volta è un dipendente e quindi a sua volta è presente nello stesso archivio.

Tralasciamo inoltre le operazioni di

Query che producono aggiornamento degli archivi, anche se tale tematica è anch'essa molto importante.

Tralasciamo altre funzioni avanzate, specie del Paradox, che permettono operazioni ancora più spregiudicate tipo la funzione SET che permette di costruire sottostrutture di appoggio in grado di snellire le operazioni o di permetterle di molto complesse.

Che ci si fa con una Query

Una volta messa a punto, una Query può essere salvata (è un file con designazione *.QbE nel dBASE, è uno Script (memorizzazione della sequenza di comandi nel Paradox) per usi successivi. Con il dBASE si può tradurre l'archivio virtuale ottenuto con la Query in un archivio a tutti gli effetti e quindi di tipo DBF. Questa operazione in Paradox è invece implicita nel senso che ogni Query produce un archivio che si chiama sempre ANSWER. Se lo si vuol conservare basta cambiargli nome, altrimenti viene sostituito non appena si esegue un'altra Query.

Il file dBASE *.QbE è comunque leggibile e comprende classici comandi dBASE eseguibili anche dall'ambiente Dot Prompt o inseribili in un programma PRG.

Le prestazioni della Query

In queste sperimentazioni non abbiamo eseguito prove cronometrate delle prestazioni dei due prodotti.

Va però detto che in dBASE IV il QbE è una sovrastruttura che comporta comunque la traduzione delle impostazioni in comandi Dot Prompt e quindi, per il semplice fatto di dover passare attraverso una traduzione, le operazioni risultano più lente.

Altro elemento a sfavore del dBASE IV è costituito dal fatto che, come noto, il dBASE IV organizza gli archivi con gli indici, per cui si possono migliorare le prestazioni delle Query aprendo anche gli indici degli archivi.

Se gli indici non vengono tenuti aperti o se questi, per quanto aperti, non servono alle operazioni richieste nella Query, la stessa o si crea dei propri indici oppure esegue lentissime operazioni sequenziali.

Il Paradox invece ha come unico linguaggio di interrogazione degli archivi il QbE, che quindi risulta più ricco e potente di quello del dBASE. Il problema delle prestazioni è risolto per mezzo di algoritmi interni di ottimizzazione che riescono a fornire risposte in tempi rapidi anche in caso di impostazioni particolarmente «incasinata».

Conclusioni

Il linguaggio QbE rappresenta effettivamente una modalità di interazione semplice ed intuitiva con gli archivi, ma al contempo sofisticata in termini di ricchezza di termini sintattici e potente in termini di risultati raggiungibili.

Meriterebbe, al pari di qualsiasi linguaggio, approfondimenti e sperimentazioni specie su problematiche pesanti.

Tra le due versioni è sicuramente più ricca e potente quella del Paradox, nel quale il QbE è senza dubbio la caratteristica fondamentale. Limitata e, in certi casi, lenta è la versione dBASE IV, che però ha dalla sua il fatto di essere solo un accessorio in un ambiente che ha ben altre funzionalità e quindi può avere una funzione propedeutica per chi deve continuare negli studi.

Ciò nonostante la strada dell'unificazione dei linguaggi è senza dubbio quella giusta.

Si è ormai imposta con lo standard dei prodotti spreadsheet, rappresentato dal Lotus 123, si può imporre, anche se non in termini così definitivi, anche nei prodotti di tipo DBMS, e questo sarebbe un grosso vantaggio per tutti gli utenti di tutte le categorie. **MC**