



## Borland dBASE V per Windows

di Francesco Petroni

**C**onfesso di essere molto affezionato al dBase, di cui ricordo ancora la versione II, nata ai tempi del CP/M (inizio anni '80). Il dBase II, che era della Ashton Tate, permetteva di aprire ben due (dico 2) tabelle contemporaneamente e di definire una relazione tra le due.

Il successo però è arrivato con il dBase III, prodotto fondamentalmente solido, adatto sia ad un utilizzo elementare, anche da parte di un utente alle prime armi che poteva sfruttare l'ambiente Assist, che ad un utilizzo evoluto, grazie sia alle sue potenti istruzioni «insiemistiche» eseguibili dall'ambiente puntino, sia al suo linguaggio di programmazione, di tipo interprete, orientato ai dati, immediatamente utilizzabile e produttivo.

Molti sono stati i motivi del successo del dBase III (per la cronaca siamo giun-

ti alla metà degli anni '80): le buone prestazioni, la facilità d'uso, l'affidabilità, il fatto di essere basato su una struttura dei file facilmente accessibile (al punto che oggi tutti i prodotti leggono e in molti casi scrivono il formato dBase) il fatto di non aver avuto, all'epoca, veri concorrenti.

Sono poi nati moltissimi altri prodotti basati sulla sintassi dBase, alcuni dei quali vivono tutt'ora. Ad esempio il Clipper, ora della C.A., o il Fox Pro, ora della Microsoft. Questi due, aggiornati ambedue per Windows (il primo si chiama Object Vision), sono ora i principali concorrenti dell'attuale dBase V.

Poi è arrivato, ancora in Ashton Tate, il dBase IV, migliorato rispetto al predecessore soprattutto nella parte interfaccia utente, con il Centro di Controllo, ma sostanzialmente invariato nella par-

te comandi. In quel periodo sono cominciate le prime difficoltà per l'Ashton Tate che, pur disponendo di due buoni prodotti (in listino c'era anche il Framework) e pur avendo avuto i due un buon successo di mercato, dopo una serie di traversie economiche è stata incorporata dalla Borland. Del dBase IV ne è uscita quindi una successiva versione marcata Borland così come del Framework.

La nuova versione, la V, è totalmente Borland e si inquadra in una sua strategia più ampia di accesso ai dati, strategia che coinvolge anche il «vecchio» Paradox e il nuovissimo Delphi, ambedue Borland, ambedue orientati all'accesso ai dati. Di Delphi proporremo la nostra prova nel prossimo numero di MC, occasione in cui parleremo anche di questa strategia.



## Il materiale e l'installazione

Precisiamo subito una cosa: il dBase V è un prodotto che si presta a svariate tipologie di utilizzo a seconda del tipo di utilizzatore (dall'utente finale al programmatore evoluto) e a seconda della tipologia di applicazione da realizzare (da una semplice attività di Personal Computing ad una complessa applicazione Client/Server in rete). La manualistica è quindi abbondantissima (un totale di oltre 2.000 pagine) e copre tutte le combinazioni di necessità.

Nei manuali ci sono comunque sezioni speciali dedicate ai «vecchi» utilizzatori che vi trovano concentrate le novità riguardanti i singoli argomenti. Ci sono anche piccole sezioni con i trucchi.

La nostra prova sarà limitata, per ora, ad una descrizione generale del prodotto e del materiale a disposizione e ad un approfondimento delle tecniche evolute d'uso del dBase, praticabili senza programmare. Ci riserviamo in un prossimo articolo di affrontare in maniera sistematica il nuovo ambiente di sviluppo, in cui entrano in gioco tecniche di programmazione Object Oriented, che merita uno spazio specifico.

La scatola con il prodotto è molto voluminosa, il che dà subito una sensazione, anche fisica, di abbondanza. All'interno troviamo i sette dischetti con il prodotto da usare per l'installazione (figura 1) e i vari manuali, che sono nell'ordine:

### dBASE V per Windows

**Produttore:**  
Borland - Francia  
**Distributore:**  
I.S.S.  
Via Valera, 31-16  
20020 Arese  
Tel.: 02/93582260  
**Prezzo (IVA esclusa):**  
dBASE V per Windows

Lit. 990.000

Figura 1 - Borland dBase V per Windows - Un momento dell'installazione.

La composizione in moduli del dBase V risulta molto chiara nella videata che presenta le opzioni di installazione. Il fabbisogno di spazio varia da un minimo di 8 mega, se si installa solo il dBase V, ad un massimo di oltre 21 mega, se si installano tutti i moduli, sia quelli operativi, come il Crystal Report, che quelli che forniscono documentazione o addestramento, come l'Help e il Tutorial.

**Getting Started**, di 70 pagine. Contiene le indicazioni per l'installazione, una prima introduzione all'ambiente operativo ed agli oggetti in esso presenti, una prima introduzione al linguaggio dBase e al linguaggio SQL. C'è anche una introduzione generale alle tecniche Windows, per chi avesse la cattiva idea di imparare ad usare dBase V pur non conoscendo ancora Windows.

**User's Guide**, di 330 pagine. È quello che parla in maniera sistematica e chiara dei vari argomenti. È quello che contiene sezioni specializzate per i «vecchi» utilizzatori del dBase. È fondamentale sia per chi inizia il suo rapporto

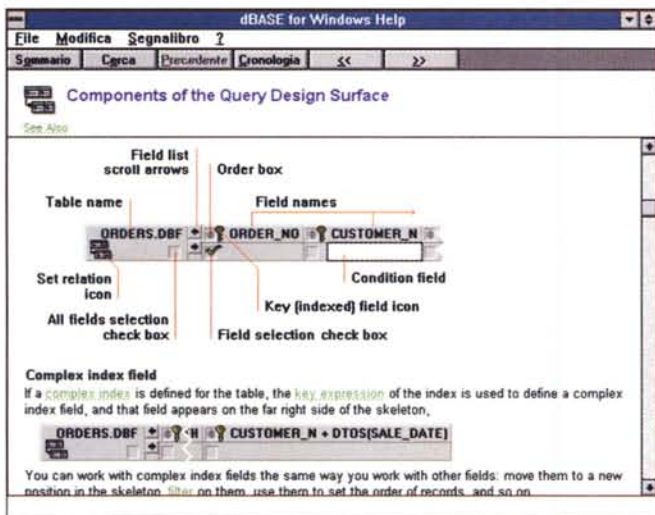
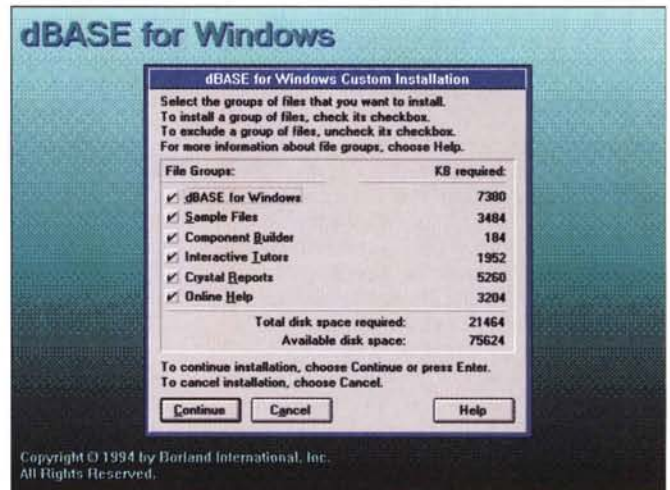


Figura 2 - Borland dBase V per Windows - Una pagina dell'help dedicata al QbE. L' Help, ovviamente alla Windows, è molto dettagliato. È soprattutto dedicato agli utenti alle prime armi. Qui vediamo una chiara descrizione dell'ambiente Query, fondamentale punto di passaggio per le applicazioni che lavorano su più tabelle.

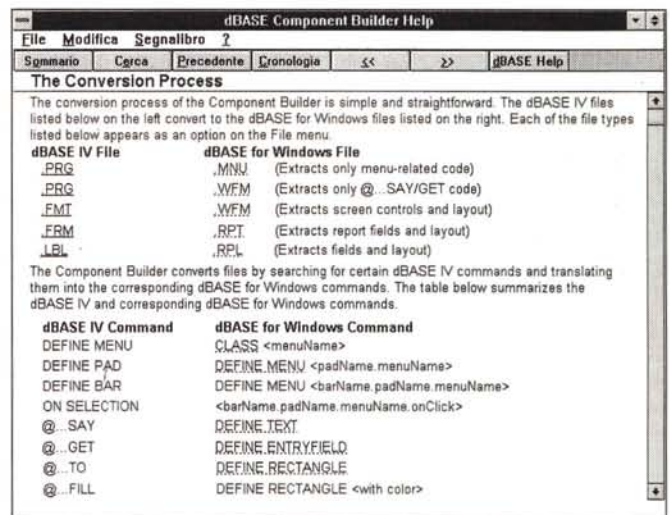


Figura 3 - Borland dBase V per Windows - Help del Component Builder.

Il problema del riutilizzo delle vecchie applicazioni scritte con il dBase, III o IV, per DOS viene risolto da un convertitore, il Component Builder, che legge i programmi, i formati, i report, ecc. scritti con la precedente sintassi e li converte nella nuova. E però chiaro che per un completo sfruttamento delle potenzialità tipiche dell'ambiente Windows occorreranno ulteriori interventi sul codice.



con il dBase sia per chi lo prosegue. Parla anche dei Tool a disposizione, come l'Espression Builder, l'Editor, la Command Window, e così via. In appendice alcuni argomenti più specifici, come le tecniche di passaggio dal dBase IV (vedi figura 3), i vari settaggi definibili a livello di DBASEWIN.INI, l'accesso ai Database SQL, ecc.

**Using Crystal Reports for dBase**, di 40 pagine. Crystal Reports è un programma accessorio che serve per costruire i Report (nelle figure e nelle didascalie ne vediamo e spieghiamo le potenzialità).

**Programmer's Guide**, di 370 pagine. Riguarda i programmatori. Il linguaggio dBase è nato con il prodotto dBase (e con il formato DBF), nel senso che le varie istruzioni utilizzabili nell'«ambiente puntino» possono anche essere inserite in un programma, che può inoltre comprendere istruzioni «in più» non eseguibili interattivamente dal puntino. Nella sua evoluzione in Windows la tradizionale struttura dBase si è coniugata con le necessità dell'Object Orientation. Il linguaggio dBase V per Windows, quindi, pur non tradendo le sue origini, diventa un linguaggio Object Oriented, in cui vengono introdotti i concetti di Oggetto, Classe, Proprietà, Evento e Metodo. Lo sviluppo del codice avviene utilizzando strumenti di programmazione visuale e, nel caso del disegno delle Form, si può utilizzare, perlomeno per smaltire il grosso del lavoro, una proce-

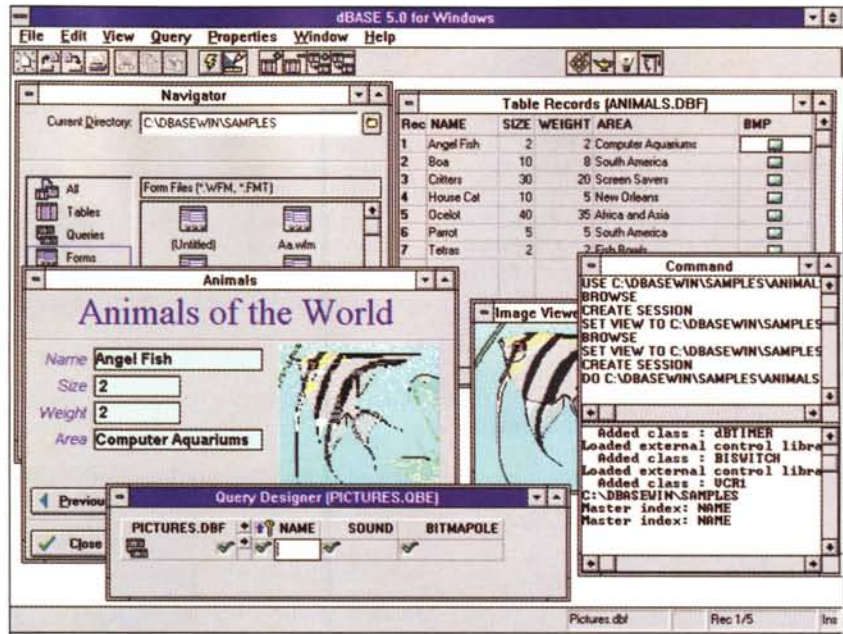


Figura 4 - Borland dBase V per Windows - Ambiente Operativo. Ogni oggetto dBase V, ad esempio una Tabella, una Query, una Form, dispone di una sua finestra posizionabile a piacere sulla videata. Poi ci sono finestre di tipo «generale», come quella che si chiama Navigator, che serve per organizzare gli oggetti di cui è composta l'Applicazione, e quella che si chiama Command che ripropone il classico ambiente puntino (Dot Prompt).

dua di Autocomposizione chiamata Form Expert. Il Programmer's Guide parla di tutte queste cose. I capitoli sono: Introduction to programming in dBase, Program Basic, Object ad Classes,

Forms, Tables, Windows environment (DDL, API, DDE). Poi ci sono le varie appendici tecniche alcune delle quali ancora una volta dedicate a chi proviene dalle versioni precedenti.

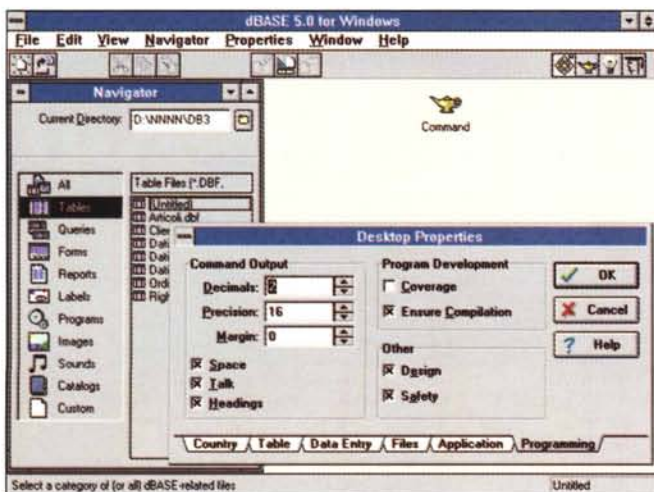


Figura 5 - Borland dBase V per Windows - Desktop Properties.

Vediamo due cose, il Navigator, ovvero il contenitore di tutti gli oggetti presenti in un'Applicazione, o meglio in una sottodirectory di lavoro, e, in primo piano, la Finestra di Dialogo Desktop Properties che consente di definire tutti i settaggi d'ambiente. È la versione Windows del comando SET.

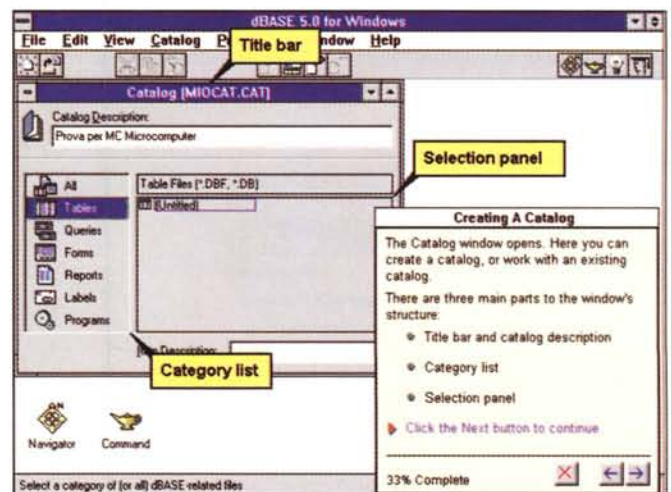


Figura 6 - Borland dBase V per Windows - Tutorial Interattivo.

Borland dBase V per Windows deve soddisfare diverse categorie di utilizzatori. Dagli più smaliziati programmatori in linguaggio dBase, ai programmatori specializzati nell'Object Oriented, ai Power Users che lo utilizzano per conto proprio sui propri dati e che vogliono «spremerli» (i dati) il più possibile. Non possono essere trascurati i neofiti per i quali dBase offre una serie di funzionalità facilmente utilizzabili. Soprattutto a costoro è destinato il Tutorial Interattivo richiamabile direttamente dal menu principale. Qui vediamo la spiegazione del concetto di Catalogo.



**Language Reference**, un malloppone di 1.050 pagine (il che porta il totale delle pagine, comprendendo nel calcolo il Quick Reference di 100 pagine, a 2.000). La parte del leone la fa l'elenco alfabetico dei comandi e delle funzioni, che sono, ovviamente, tantissime, dato che comprendono anche le estensioni dovute all'introduzione degli Oggetti e delle Classi. Altri due capitoli importanti sono quello che parla in maniera sistematica di Classi e quello che parla di Properties. È chiaro che nella descrizione di ogni classe di oggetti c'è l'elenco delle proprietà gestibili e, viceversa, nella descrizione della singola proprietà c'è l'elenco delle classi cui si può applicare.

L'ultimo manuale è il **Quick Reference** che è un sunto, con comandi, funzioni, classi e proprietà, del Reference. È di dimensioni più piccole rispetto agli altri per facilitarne la consultazione.

**L'ambiente**

Una volta installato e lanciato il dBase V appare il suo ambiente operativo (fig. 4) organizzato in maniera abbastanza tradizionale, con una pulsantiera in alto, sotto il menu, e la finestra Navigator. Sia la Navigator che le altre finestre con i vari elementi sono finestre MDI. Ogni finestra può essere massimizzata, oppure minimizzata, oppure piazzata a piacere all'interno della finestra esterna. L'elemento principale è il Navigator, ovvero il contenitore dei vari oggetti, che

vengono suddivisi per tipo. Il Navigator «punta» verso una sottodirectory che può essere considerata il contenitore «fisico» dei vari elementi costituenti l'applicazione. Un'altra forma di organizzazione è quella costituita dal Catalogo, in cui possono essere inseriti elementi comunque distribuiti sull'hardisk (fig. 6).

La finestra Navigator mostra in alto la directory puntata e poi, sulla sinistra, le icone che rappresentano i vari tipi di oggetto. Ci sono Tables, Query, Form, Reports, Labels, Programs, Images, Sounds, Catalogs e Custom, in alto l'Icona All. Altra finestra di tipo generale è quella Command, che mostra una zona nella quale si possono digitare direttamente i comandi «puntino» e una zona sottostante nella quale si concretizza l'effetto del comando puntino. Se in alto si scrive:

**USE CLIENTI LIST**

Nella parte inferiori scorre la lista dei Clienti.

Se, al contrario, si lavora con i menu nei vari ambienti interattivi, nella Command Window ritroviamo i corrispondenti comandi puntino.

In alto a destra, nella Toolbar principale, ci sono i quattro pulsanti per richiamare il Navigator (c'è anche un film di Buster Keaton che si chiama così), la Command Window, il Form Expert e il Tutorial Interattivo (fig. 6).

Nella finestra Navigator sono elencati quindi i vari tipi di oggetto. Tables,

Query, Form sono i più importanti. Poi ci sono Report e Label, ambedue richiamano il Crystal Report, poi Images e Sounds, che richiamano i programmi, in genere PaintBrush e il Media Recorder di Windows, con i quali editare i due tipi di oggetto usabili nei campi di tipo OLE.

Vediamo ora un po' più nel dettaglio i principali tipi di elemento.

**Le Tabelle**

In dBase V viene utilizzata una nuova terminologia per definire i vari elementi. Ora esiste il Database, inteso come collezione di una o più Tabelle e di file correlati alle tabelle, come Indici e file Memo. Prima invece si usava il termine Database per indicare (scorrettamente) le Tabelle.

Il contenitore fondamentale dei dati è quindi la Tabella che coincide con il classico file DBF. Gli indici risiedono o tutti insieme nel file MDX, oppure, ognuno per conto suo, nei file NDX. I campi Memo risiedono nei file DBT. I nuovi tipi di campo disponibili in dBaseV sono: campo di tipo binario, per memorizzare immagini o suoni, e campi di tipo OLE. Vanno a finire nel file DBT. Questo rende impossibile l'utilizzo di un file DBF con campi di tipo OLE, creato quindi con dBase V per Windows, da una versione precedente di dBase. C'è anche il campo di tipo Float che serve per memorizzare numeri, o numeri piccolissimi o numeri grandissimi, da espri-

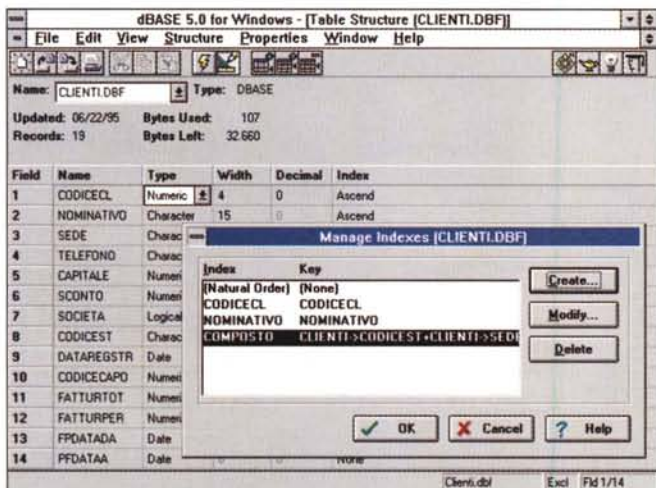
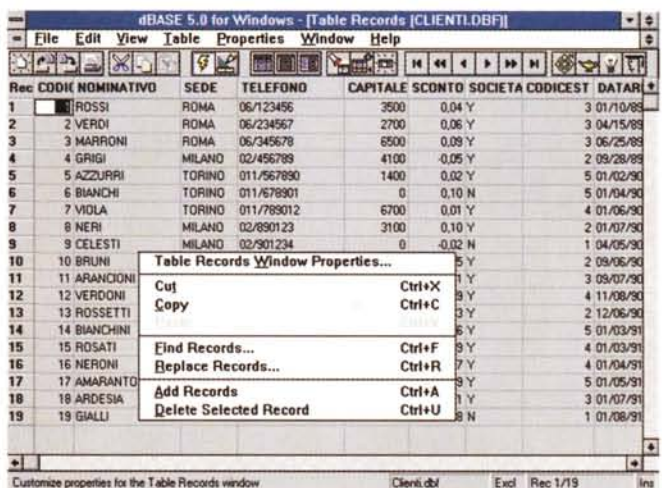


Figura 7 - Borland dBase V per Windows - La struttura della tabella e gli indici.

Vediamo una fase della creazione della struttura del file DBF. La struttura tradizionale è stata modificata per poter inserire alcune nuove tipologie di campo, necessarie in Windows. È chiaro che questo fatto rende il formato dei nuovi file DBF incompatibile verso le versioni precedenti. In primo piano vediamo la Box che serve per generare gli Indici.

Figura 8 - Borland dBase V per Windows - Browsing della Tabella.

Una Tabella può essere vista in varie maniere. In formato «colonnare», con un campo per riga, in formato Form, con i campi, nome e contenuto, impacchettati, in formato tabellare, che è quello che corrisponde al classico Browse. In ogni caso sono possibili interventi di tipo organizzativo ed estetico. Ad esempio è possibile scegliere il tipo di carattere oppure l'ordine delle colonne nel Browser.





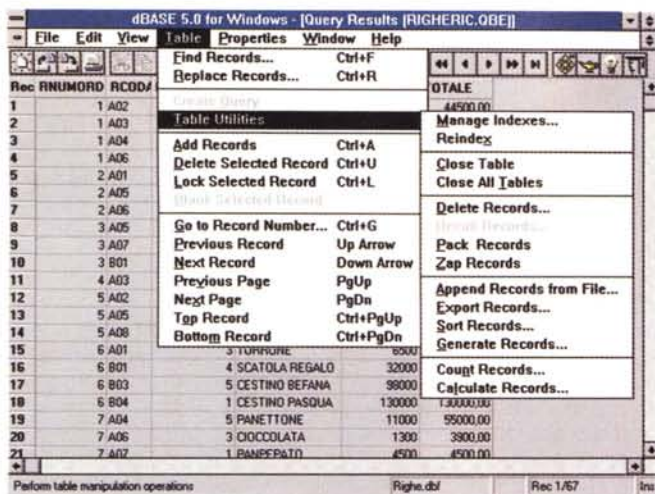


Figura 9 - Borland dBase V per Windows - Cosa ci si fa con una Tabella o una Query. Rispetto alle possibilità proprie degli Assist e dei Control Centre delle versioni DOS, la versione per Windows del dBase presenta numerosi vantaggi operativi, non tanto per le ovvie possibilità «estetiche», quanto per il fatto che le varie operazioni manipolative sulle Tabelle o sulle Query (è il caso che vi stiamo mostrando) sono immediatamente attivabili o dalle Toolbar o dai menu.

Creata la tabella ci si può direttamente lavorare (fig. 8). Le modalità operative, immediatamente attivabili, sono tre: quella con il Layout a Colonna, con quello a Form e con quello Tabellare. Più in là vedremo come creare le Form per ottimizzare le sessioni di Data Entry.

La Toolbar è molto ricca di pulsanti e sono molti abbondanti le voci di menu che eseguono operazioni sulla Tabella attiva. Ad esempio si può stampare direttamente il contenuto della Tabella, secondo il Layout che appare sul video, oppure si può switchare sulla struttura, se si deve modificare qualche cosa. La pulsantiera VCR-like ha sei tasti per muoversi sul primo record, sulla pagina precedente, sul record precedente, sul successivo, e così via. Sono anche attive una serie di funzionalità «estetiche» ed organizzative.

Se si lavora sulla Tabella, ma il discorso vale anche per i vari altri tipi di oggetto, è disponibile una Box che si chiama Table Records Property, che serve per specificare al meglio le modalità di lavoro sulla tabella stessa, ad esempio quali campi vedere, in quale ordine, se limitare i record, ecc.

Già a livello di Tabella è possibile definire altri elementi e caratteristiche dei campi. Ad esempio regole di validazione, formati di visualizzazione, ed altro.

Sui dati delle Tabelle, qualsiasi sia la forma di visualizzazione scelta, si può intervenire con operazioni di tipo indivi-

mere quindi in notazione scientifica. Anche questi prima non c'erano.

La struttura della Tabella si crea in uno speciale Editor abbastanza Windowsizzato (figura 7), che permette alcune operazioni di Drag and Drop, su cui si agisce con il tasto destro del mouse, e così via. Gli indici possono essere di due tipi: o indici associati al campo, oppure indici derivati da un'espressione, ad esempio un indice calcolato, oppure un indice composto. Occorre utilizzare una Dialog Box nella quale si scrive o si compone l'espressione.

In molte situazioni operative, quando occorre inserire delle formule, si può attivare il generatore di formule (si richiama con un pulsante che mostra una chiave inglese) che vediamo in una successiva figura.

Il concetto di indice è fondamentale in un prodotto di tipo DBMS e i vari manuali lo trattano con il dovuto approfondimento. Nei manuali di dBase V vengano, ad esempio, trattate le differenze tra il Sort e l'Indice, oppure vengono suggerite le modalità d'uso degli Indici nelle operazioni di Data Entry, oppure ancora nelle Query o nei Report.

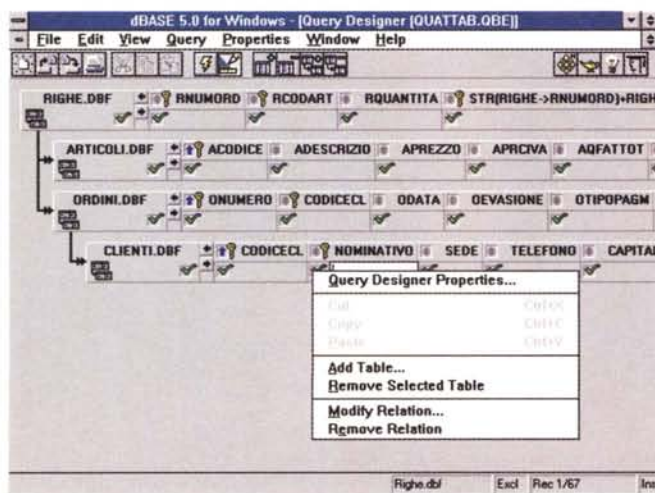


Figura 10 - Borland dBase V per Windows - Costruzione di una Query. La costruzione della Query avviene in questo ambiente Query by Example, in cui appaiono le strutture delle Tabelle coinvolte, le linee che indicano le Relazioni tra le Tabelle, altri simboletti che indicano se e come i campi debbano apparire in uscita. Le Query sono viste dirette sui dati. In altre parole è possibile aggiornare i dati direttamente dalla Query.

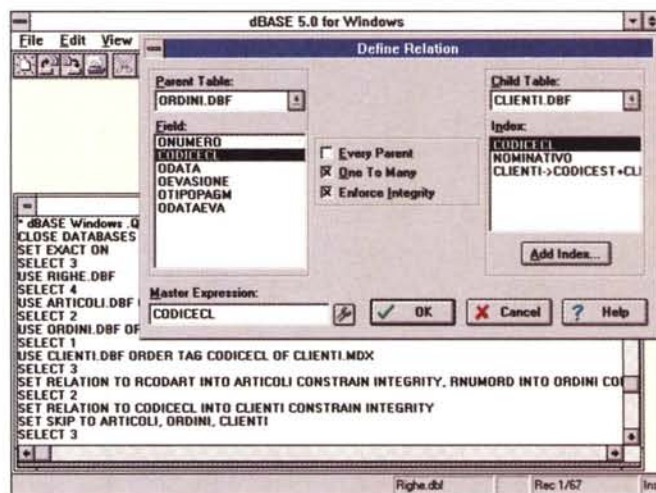


Figura 11 - Borland dBase V per Windows - La definizione delle Relazioni.

In questa figura vediamo due cose. La Finestra Comandi (corrispondente all'antico ambiente puntino) con tutti i comandi che sono stati «registrati automaticamente» quando abbiamo creato le tre relazioni tra le quattro tabelle usate per la prova e, in primo piano, la Box con la quale si definiscono, in maniera assistita, le relazioni e i controlli nelle relazioni. Fondamentale il fatto che il dBase V assume ora su di sé l'incombenza di controllare l'integrità referenziale dei dati.



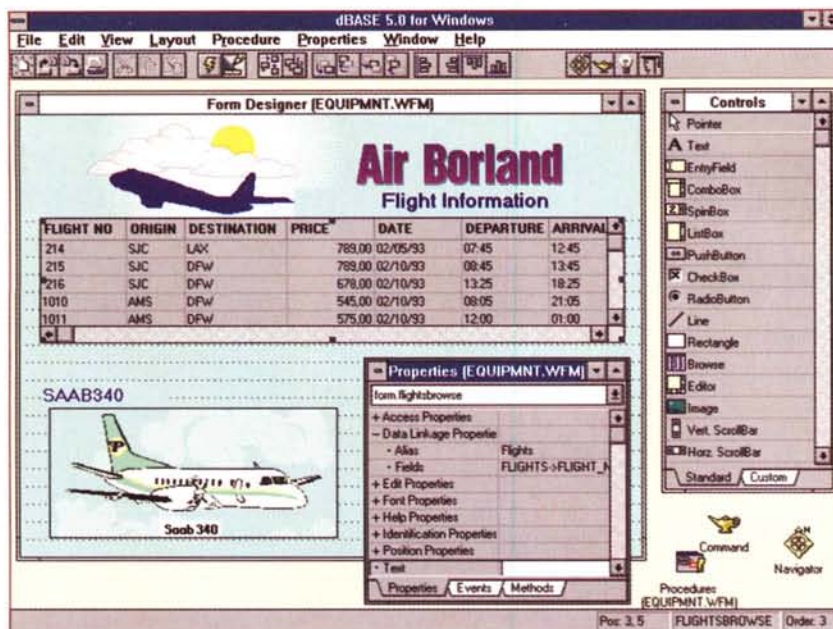


Figura 12 - Borland dBase V per Windows - Ambiente Disegno Form. Anche in fase di costruzione della Form appaiono i dati sottostanti. Questo permette un controllo immediato della correttezza delle impostazioni eseguite. In fase di disegno si usano anche altre due finestre, quella Controls, che propone tutti i Controls inseribili in una Form, e quella Properties, che propone tutte le caratteristiche dell'oggetto su cui si sta lavorando. Si tratta di una classica programmazione Object Based.

vanno definite le Regole Relazionali tra la Tabelle, il secondo è che le Query non equivalgono a delle semplici estrazioni di dati ma sono delle viste dirette sui dati, che sono quindi aggiornabili direttamente dalle Query. Questo consentirà anche, e lo vedremo tra un po', di realizzare Form e Report sui dati di una Query e non semplicemente sui dati di una Tabella.

Le Query, che potremo definire Interrogazioni sui Dati, si costruiscono in uno speciale ambiente grafico, evoluzione di quello già visto nel dBase IV, di tipo QbE (Query by Example), in cui si può specificare, sfruttando la strumentazione grafica disponibile:

- quali tabelle usare (comando dBase USE),
- quali sono le relazioni tra le tabelle (comando dBase SET RELATION),
- quali campi di visualizzare, con eventuali campi calcolati (SET FIELDS),
- quali record visualizzare (SET FILTER),
- in quale ordine visualizzarli (SET INDEX, INDEX, SORT, ecc.).

Alla fine viene prodotto un file \*.QBE in cui viene memorizzata la «regola» di estrazione e non il suo risultato. Quindi, una volta che la Query stessa viene richiamata, viene automaticamente rieseguita e i dati che mostra sono quelli aggiornati.

Nell'ambiente operativo, inizialmente vuoto, si inseriscono le varie tabelle da cui prelevare i dati. Le Tabelle appaiono sotto forma di Struttura e mostrano

duale, sul singolo record quindi, o con operazioni insiemistiche. In quest'ultimo caso tutte le specifiche per la ricerca dei dati desiderati possono essere impostate in opportune Dialog Box (fig. 9).

In conclusione anche il semplice lavoro su una Tabella può essere bene organizzato e controllato, senza dover ne-

cessariamente scomodare i successivi tipi di oggetto.

### Le Query

In dBase V le Query assumono un'importanza «strategica» per vari motivi. Il primo è che è a livello di Query che

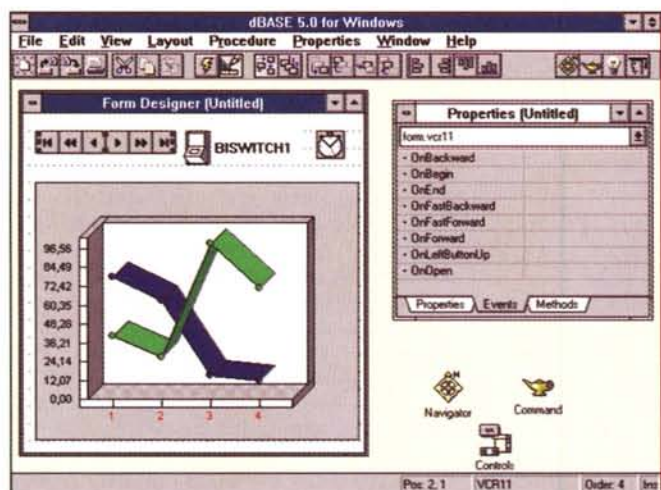


Figura 13 - Borland dBase V per Windows - Custom VBX.

Quando si costruisce una Form si scelgono i controlli da inserire nella Form stessa da una specifica finestra. Questa mostra quelli standard e quelli Custom. Questi ultimi possono essere o pulsanti predefiniti, con tanto di codice incorporato, e possono essere dei control VBX, compatibili con quelli di Visual Basic e di Delphi. Qui vediamo quelli in dotazione: c'è un Timer, un Interruttore, una Pulsantiera VCR e un Generatore di Grafici Business molto evoluto.

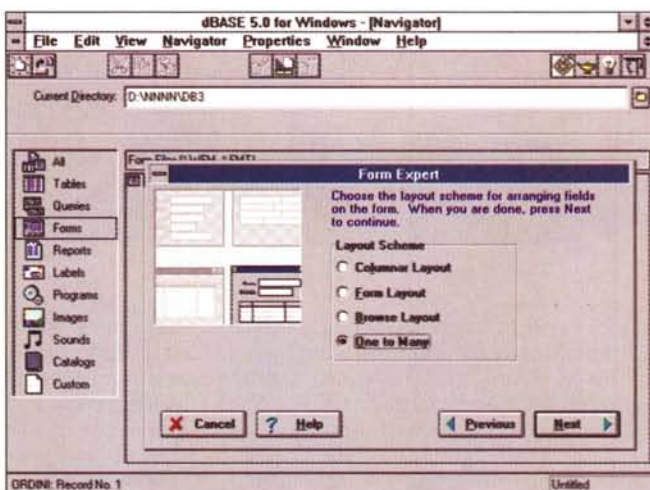


Figura 14 - Borland dBase V per Windows - La costruzione della Form con il Form Expert.

Per costruire la Form si può utilizzare una procedura automatica, chiamata Form Expert, che permette di costruire Form di tipo Colonnare, di tipo Scheda (Form), di tipo Tabellare (Browse) e di tipo Uno a Molti. In questo ultimo caso c'è una zona superiore, in formato Colonnare, che mostra i dati lato Uno della relazione (nel nostro caso vedremo i dati degli Ordini e i dati del Cliente) e una zona inferiore, in formato Browse (che mostra i dati delle Righe e degli Articoli).



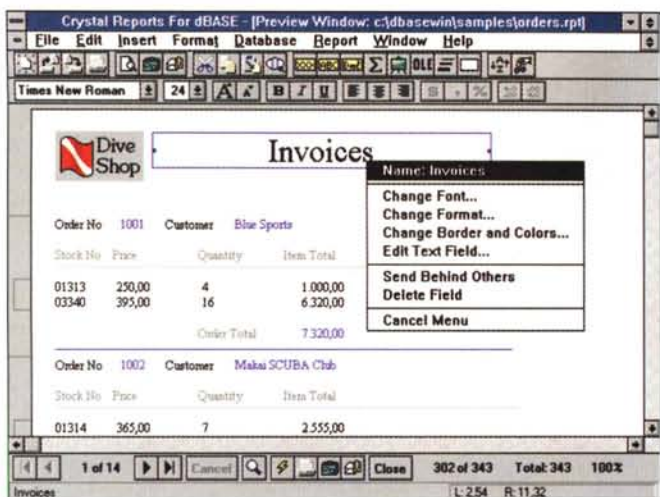


Figura 15 - Borland dBase V per Windows - Crystal Report - Costruzione.

dBase V si appoggia per la produzione dei Report su un prodotto esterno, il Crystal Reports, già inserito allo stesso scopo nel Visual Basic della Microsoft. La versione per dBase V, che peraltro lavora gomito a gomito con dBase stesso, permette sia impostazioni di tipo contenutistico che impostazioni di tipo estetico. Qui vediamo un Report organizzato su due livelli di raggruppamento.

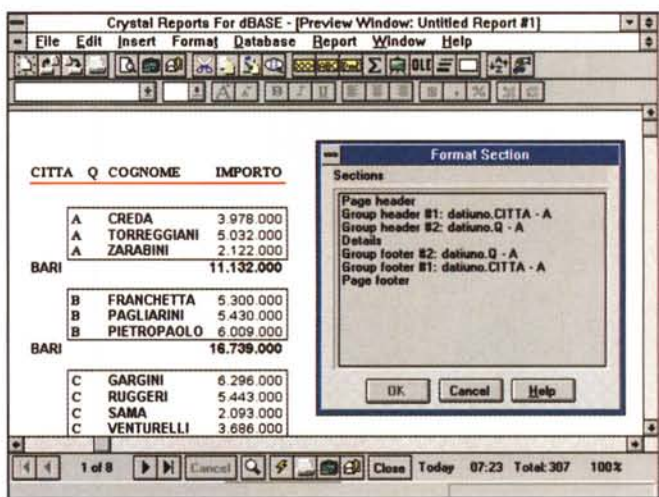


Figura 16 - Borland dBase V per Windows - Crystal Report - Campo Calcolato.

In questo caso vediamo come sia possibile aggiungere, al Report, dei campi calcolati. In pratica viene sfruttato il generatore di espressioni del dBase V che dispone anche della comoda possibilità di testare il buon funzionamento dell'espressione composta. Interessante è il fatto che anche in fase di costruzione si vedano i dati veri.

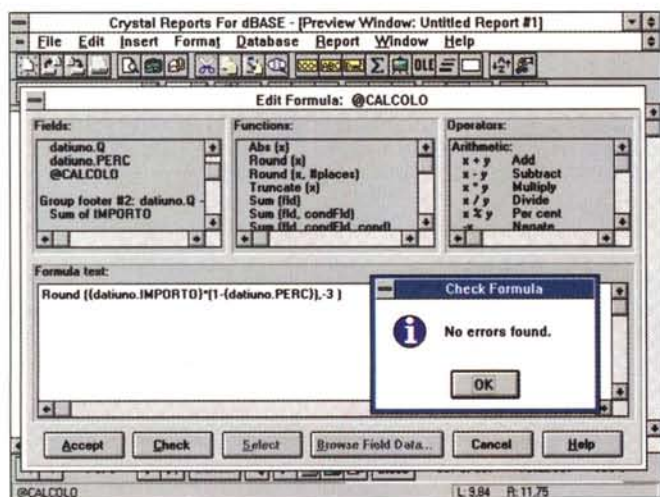


Figura 17 - Borland dBase V per Windows - Crystal Report - Report Multitabella.

Un Report può appoggiarsi anche su una Query e quindi può mostrare dati provenienti da più tabelle. Questa possibilità allarga gli ambiti di utilizzo del Crystal Report anche a problematiche un po' più spinte. In questo caso vediamo degli Ordini, al di sotto dei quali sono state disposte le righe di Dettaglio. Si tratta di una classica problematica relazionale.

vece l'ordine d'inserimento è Ordini, Clienti e Dettagli (e magari Articoli) si produce un elenco di Ordini con indicazione del Cliente (la relazione viene percorsa nell'altro senso) e con le varie Righe sottostanti con i Dettagli, arricchiti con i dati degli Articoli. Questa situazione è quella che si dovrà usare per predisporre una Form Ordini con tutte le specifiche dei Dettagli delle Righe (è quello che faremo poi).

La Box nella quale si definiscono le relazioni non mostra l'indicazione Uno o l'indicazione Molti, ma l'indicazione Parent e quella Child (che possono essere, a seconda dei casi, il lato Uno o il lato Molti della Relazione). Quindi, nel predisporre la Relazione tra due tabelle, occorre indicare il campo della tabella «lato Parent» e l'indice della tabella «lato Child» della relazione, e questo indice deve essere costruito (si può fare lì per lì) sul campo della tabella Child che corrisponde a quello indicato come campo di collegamento nella tabella lato Parent.

Nella Box per la definizione delle Relazioni sono presenti tre switch-etti:

- Every Parent, per visualizzare anche i record lato Parent che non hanno corrispondenti lato Child (esempio elenco di Clienti con i rispettivi Ordini che comprenda anche i Clienti che non hanno Ordini). Comando corrispondente: SET FILTER TO FOUND().
- One to Many, significa che ad un record della tabella Parent corrispondono uno o più record nella tabella Child.

quindi i nomi dei campi e una serie di bottoncini che servono per scegliere i campi da visualizzare, le regole di ordinamento, i criteri di estrazione (fig. 10). Tra le Tabelle si possono «tirare» delle linee che costituiscono le Relazioni: appare una Box nella quale si stabiliscono le proprietà della Relazione. In figura 11 vediamo la Box e vediamo, in secondo piano, la Command Window che mostra i vari comandi «puntino» di apertura tabelle e definizione relazioni.

Abbiamo appena detto che è nella Query che vanno stabilite le regole relazionali che poi vengono controllate nelle

successive operazioni. Ad esempio, se si debbono raggruppare dati provenienti da più tabelle per usarli in un Report o in una Form, si prepara una Query che le collega. L'ordine ed il modo con cui sono collegate incide sul modo con cui «escono» i dati e quindi con il modo con cui questi possono essere utilizzati.

Ad esempio in una situazione con tre tabelle: Clienti, Ordini e Righe di Dettaglio (cui corrispondono due relazioni Uno a Molti) le stesse si possono inserire nella Query in questo stesso ordine, ad esempio per un Report che mostri Clienti con Ordini e con i Dettagli. Se in-



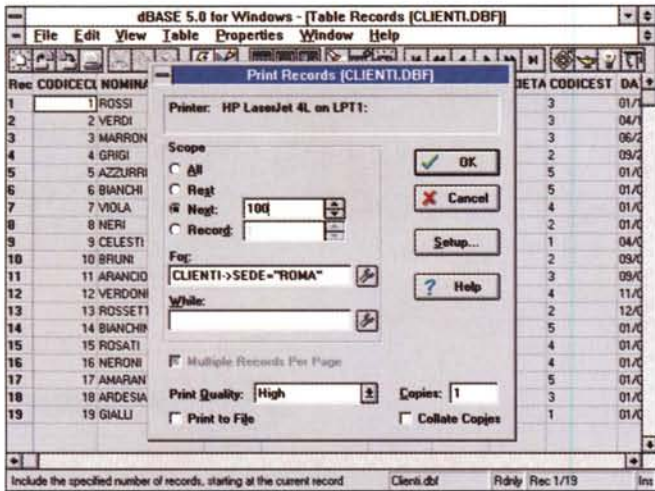
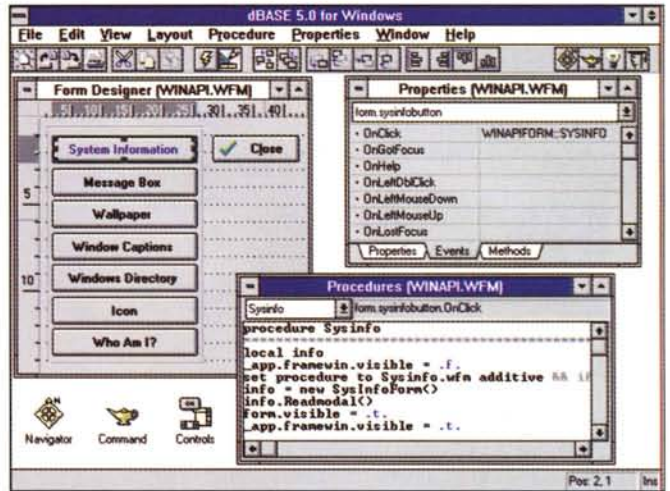


Figura 18 - Borland dBase V per Windows - Stampa diretta.

Da tutti gli ambienti in cui si vedono dati è possibile attivare una loro stampa diretta. Ad esempio se si parte dalla Tabella, vista in formato Browse, e si fa click sul pulsante di stampa si apre una Box nella quale si possono impostare le varie clausole di stampa ed in particolare i «filtri» con i quali si selezionano i dati.

Figura 19 - Borland dBase V per Windows - Programmazione degli Oggetti - Visuale.

Come spiegato nel testo, in questa prova non arriveremo a parlare, per mancanza di spazio, della programmazione con dBase V, che sarà tema di un articolo specifico tra due numeri di MC. Vi anticipiamo solo gli aspetti fondamentali e cioè che la tradizionale sintassi del linguaggio dBase si è convertita alla Object Orientation, e che alla programmazione tradizionale si può sovrapporre, per quanto riguarda lo sviluppo delle Form, che costituiscono il cuore dell'applicazione da sviluppare, la programmazione Visuale.



Sempre nel caso di prima è la specifica che serve per vedere tutti gli Ordini di ciascun Cliente e non solo il primo Ordine. Comando corrispondente SET SKIP.

- Enforce Integrity, significa che non si può cancellare un record Parent se esiste un record Child, né inserire un Child se non esiste un Parent corrispondente. Il comando relativo è SET RELATION .. INTEGRITY. Quando si cerca di cancellare un record lato Parent dBase V richiede se si vogliono cancellare «a caduta» anche tutti i Child corrispondenti.

Insomma un modo un po' personale di vedere le relazioni, ma completo ed efficace.

Mi scuso della «tirata» sulle Query ma ritengo che la padronanza di questi «oggetti» sia fondamentale per il corretto utilizzo di un qualsiasi prodotto DBMS, in particolare del dBase V, in cui, essendo le Relazioni impostabili a livello di Query, diventano fondamentali anche per il controllo della correttezza e dell'integrità dei dati.

### Le Form

Le Query sono, come appena visto, lo strumento fondamentale per la predisposizione dei dati che vengono letti dalle Tabelle e, secondo le varie impostazioni date nella Query stessa, preparati per i successivi utilizzi. Le Form sono invece lo strumento fondamentale di interfaccia tra dati ed utente.

I dati gestiti in una Form possono

conseguentemente provenire o da una Tabella o da una Query (quindi tramite queste da più Tabelle variamente interconnesse tra di loro).

La Form, totalmente rispondente alle regole Windows, si costruisce in un Ambiente Visuale (lo vediamo in figura 12), molto simile a quello proposto dal Visual Basic della Microsoft e dal Delphi della Borland (che vedremo nel prossimo numero). I tre prodotti sono anche accomunati dal fatto che possono, tutti e tre, utilizzare i famosi Controls VBX (fig. 13).

L'ambiente nel quale di disegnano le Form è caratterizzato oltre che dalla presenza della Form «in lavorazione», dalla presenza della Toolbar, della Finestra con i vari Controls disponibili (corrispondenti a tutti i tipi di oggetto Windows), della Finestra Properties che propone le proprietà specifiche di ciascun oggetto e dalla Finestra con le procedure.

Quando si predispone una Form si possono comunque sfruttare i servizi del Form Expert, uno speciale «auto-compositore» che chiede su quale Tabella o Query lavorare e predispone una Form già operativa su quei dati (fig. 14). In questo semplice caso ci si può poi muovere sui dati usando una pulsantiera VCR-Like posta in alto nella Toolbar.

Sul disegno «automatico» della Form si può poi intervenire per modifiche e personalizzazioni, così come si può designare una nuova Form totalmente a

mano. Il processo è quello tipico della programmazione visuale:

- si inserisce un nuovo elemento, ad esempio una Casella di Testo (in dBase si chiama EntryField) oppure un Pulsante,
- se ne definiscono i legami con i dati, e cioè quale campo alimenta la casella di testo (e viceversa),
- se ne definiscono le caratteristiche estetiche,
- si definiscono le varie routine richiamate al verificarsi degli eventi, ad esempio in entrata nella Casella di Testo, in uscita, oppure al Click sul pulsante, se l'oggetto è un Pulsante.

I tipi di oggetto presenti nella Finestra dei Controlli sono tutti quelli classici di Windows, più alcuni tipi speciali già preconfezionati (ad esempio sono disponibili una serie di tasti standard in un'applicazione gestionale), più quelli caricati (si fa o al volo o definitivamente intervenendo sul file INI) come VBX.

La finestra delle Properties è molto ricca. Le Properties sono suddivise in tre pannelli differenti visualizzabili tramite le linguette e poi le singole Properties sono organizzate in maniera gerarchica, in modo che l'elenco possa essere esploso in caso di necessità oppure ridotto.

C'è poi la finestra in cui si scrive, con il mini Editor di bordo, il codice legato al verificarsi di un certo evento su un certo oggetto.

Legato alla singola Form ci può esse-



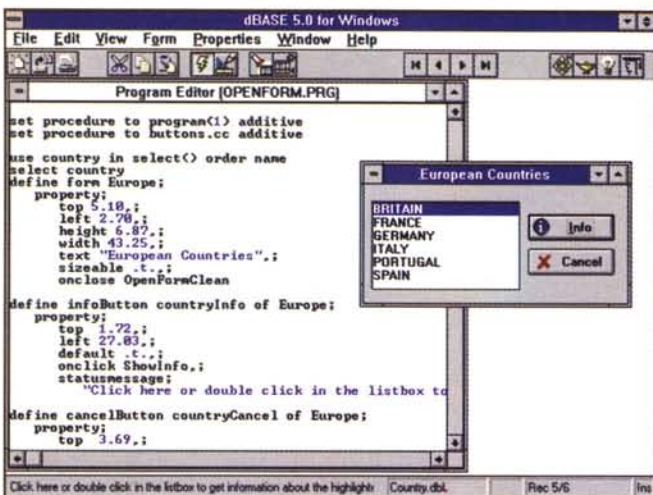
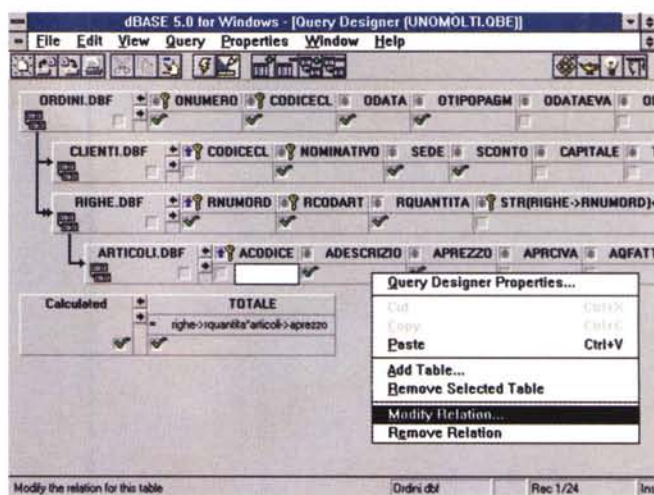


Figura 20 - Borland dBase V per Windows - Programmazione degli Oggetti - Codice.

Vediamo, a titolo di esempio, le due facce della medaglia programma. Una semplice Form, disegnabile in modo visuale ed una porzione del codice corrispondente. Nel Borland Delphi, che vedremo nel prossimo numero di MC, questi due metodi operativi paralleli si chiamano Two-Way-Tools.

Figura 21 - Borland dBase V per Windows - L'interno della Query Quattro Tabelle.

Il caso che percorriamo è il classico Clienti, Ordini, Dettagli ed Articoli. L'obiettivo è quello di costruire un'unica Form, relativa agli Ordini, che mostri ed organizzi tutti i dati insieme. Qui vediamo la struttura della Query che parte dagli Ordini e che collega questa tabella, direttamente o indirettamente, alle altre tre. L'aspetto più critico è proprio quello della definizione «fine» delle Relazioni.



re un Menu personalizzato che può essere disegnato con lo specifico generatore Menu Designer. Il processo è totalmente guidato e, alla fine, ad ogni voce di menu deve corrispondere un sottoprogramma e una chiamata all'help (gli eventi propri di una voce di menu sono On Click e On Help).

Altro Tool che entra frequentemente in gioco è l'Expression Builder la cui funzione è quella di guidare l'utente quando debba scrivere una formula di calcolo complessa. Propone le tabelle e i rispettivi campi, le funzioni di sistema (e le eventuali funzioni personalizzate), gli operatori matematici e booleani, ecc. Un tasto Evaluate serve per testare la correttezza della espressione.

Per chi ha esperienza di Visual Basic le analogie che si trovano con l'ambiente visuale dBase V per il disegno delle Form sono tantissime. Chi invece non ha nessuna esperienza di programmazione visuale deve inizialmente capire i suoi concetti di base, per poter poi operare velocemente.

### I Report e le Labels

Non esiste più il generatore di Report interno, in quanto ora si usa il Crystal Reports, che invece è un applicativo esterno, realizzato dalla Crystal Computer Services, che lo vende sia come prodotto indipendente, sia come accessorio al servizio di altri prodotti. Ne è ben nota la versione per MS Visual Basic 3.0.

Quello per dBase V è ben integrato nel prodotto al punto che lo si lancia dal Navigator e che, se lo si lancia da solo, comunque carica il dBase dal quale, evidentemente, preleva dei dati. Permette di costruire Reports di una certa complessità, con gruppi, sottogruppi, con vari livelli di totalizzazione, con possibilità di inserire righe di dettaglio (in presenza di una relazione uno a molti), con possibilità di inserire immagini fisse (ad esempio loghi aziendali) oppure campi di tipo OLE con immagini. Vediamo e leggiamo le didascalie delle figure dalla 15 alla 17.

In ogni caso è possibile, da varie situazioni, eseguire una stampa diretta. Esiste un apposito pulsante sulla Toolbar che attiva una finestra (fig. 18) nella quale possono essere impostate le specifiche di stampa, comprese le eventuali regole per la selezione dei dati da stampare. Manca, ed è un peccato, l'antepprima di stampa che permetterebbe il controllo preventivo dei dati in uscita.

### La programmazione

In dBase V convivono due tipi di programmazione, quella tradizionale basata su listati di codice e quella visuale «object based». È evidente che un utilizzatore evoluto deve conoscerle entrambe. Chi viene dal dBase classico si deve riconvertire alla Object Orientation e chi invece già mastica quest'ultima materia non deve ignorare i principi fon-

damentali del dBase. Il nuovo linguaggio è comunque compatibile con le precedenti versioni III e IV, per DOS. È chiaro che i vecchi programmi girano in una finestra Windows molto semplificata e che i vari elementi definiti sono tutti uguali. In Windows non hanno più senso alcune vecchie istruzioni, che però sono state lasciate per garantire il funzionamento dei vecchi programmi. Ad esempio la vecchia istruzione @ R,C è praticamente sostituita dalla Casella di Testo, che in dBase si chiama Entry-Field, che ha una serie di Properties, tra cui la posizione sul video.

Le novità relative alla Object Orientation riguardano la possibilità di maneggiare oggetti e classi, la possibilità di creare proprie nuove classi, e quindi porzioni riutilizzabili di codice. La programmazione è Event Driven. I vari pezzi di codice sono quindi legati al verificarsi dei vari eventi, e non vengono eseguiti in una sequenza rigorosa (in figura 19 vediamo un momento della programmazione di un evento e nella figura 20 vediamo l'Editor tradizionale).

L'ambiente programmazione dispone di una serie di funzionalità di servizio e di una serie di ulteriori «features». La più importante tra le prime è il Debugger, che può essere usato dal dentro ma anche dal fuori rispetto a dBase V. dBase V supporta Tabelle SQL, accettando per queste sia comandi dBase che comandi SQL. Sono eseguibili anche procedure «pass-through» tramite



Rec	ONU	COD	DATA	DTI	POPE	NOMINAT	SEDE	SCONT	RIG	RCD	RIGL	ADESCRIZIO	APREZZ	TOTALE
1	1	02/01/93	ASSEGNO	BIANCHI	TORINO	0,10	1	A02	5	SPUMANTE	2500	44500,0		
1	1	02/01/93	ASSEGNO	BIANCHI	TORINO	0,10	1	A03	4	GIANDUA	3600	14400,0		
1	1	02/01/93	ASSEGNO	BIANCHI	TORINO	0,10	1	A04	1	PANETTONE	11000	11000,0		
1	1	02/01/93	ASSEGNO	BIANCHI	TORINO	0,10	1	A05	3	CIOCCOLATA	1300	3900,0		
2	2	3	02/07/93	BONIFICO	MARRONI	ROMA	0,05	2	A01	5	TORRONE	6500	32500,0	
2	2	3	02/07/93	BONIFICO	MARRONI	ROMA	0,05	2	A05	10	PANFORTE	2700	27000,0	
2	2	3	02/07/93	BONIFICO	MARRONI	ROMA	0,05	2	A06	2	CIOCCOLATA	1300	2600,0	
3	3	4	02/05/92	CONTANTI	GRIGI	MILANO	-0,05	3	A05	3	PANFORTE	2700	8100,0	
3	3	4	02/05/92	CONTANTI	GRIGI	MILANO	-0,05	3	A07	5	PANPEPATO	4500	22500,0	
3	3	4	02/05/92	CONTANTI	GRIGI	MILANO	-0,05	3	B01	1	SCATOLA REGALI	32000	32000,0	
4	4	8	02/15/93	BONIFICO	NERI	MILANO	0,10	4	A03	6	GIANDUA	3600	21600,0	
5	5	10	02/18/93	BONIFICO	BRUNI	ROMA	0,15	5	A02	3	SPUMANTE	8500	25700,0	
5	5	10	02/18/93	BONIFICO	BRUNI	ROMA	0,15	5	A05	8	PANFORTE	2700	21600,0	
5	5	10	02/18/93	BONIFICO	BRUNI	ROMA	0,15	5	A08	2	CHAMPAGNE	37000	74000,0	
6	6	1	02/18/93	CONTANTI	ROSSI	ROMA	0,04	6	A01	3	TORRONE	6500	19500,0	
6	6	1	02/18/93	CONTANTI	ROSSI	ROMA	0,04	6	B01	4	SCATOLA REGALI	32000	128000,0	
6	6	1	02/18/93	CONTANTI	ROSSI	ROMA	0,04	6	B03	5	CESTINO BEFANI	56000	490000,0	
6	6	1	02/18/93	CONTANTI	ROSSI	ROMA	0,04	6	B04	1	CESTINO FRSQUI	130000	130000,0	
7	7	5	02/18/93	ASSEGNO	AZZURRI	TORINO	0,02	7	A04	5	PANETTONE	11000	55000,0	
7	7	5	02/18/93	ASSEGNO	AZZURRI	TORINO	0,02	7	A06	3	CIOCCOLATA	1300	3900,0	
7	7	5	02/18/93	ASSEGNO	AZZURRI	TORINO	0,02	7	A07	1	PANPEPATO	4500	4500,0	

Figura 22 - Borland dBase V per Windows - Output della Query Quattro Tabelle.

È chiaro che essendo partiti dalla Tabella Ordini vediamo per ogni Ordine (a sinistra) più Righe di Dettaglio. Inoltre per ogni Ordine vediamo alcuni dati del Cliente e per ogni Riga alcuni dati dell'Articolo. Partiremo da questa Query per costruire la Form.

Figura 23 - Borland dBase V per Windows - La Form Quattro Tabelle.

Il risultato dell'autocomposizione è una Form già funzionante, sulla quale si può intervenire con una serie di operazioni di tipo «estetico», di tipo organizzativo e di tipo funzionale.

la funzione SQLEXEC(). Il linguaggio dBase dispone anche di un preprocessore C-style, con il quale realizzare pezzi di codice che collaborino con il programma dBase.

A queste «features» vanno aggiunte le funzioni e i comandi che permettono di sfruttare tutti i servizi di Windows, come DLL, API, Printer e Font support, OLE, DDE, servizi multimediali.

Abbiamo preannunciato che parleremo in un altro e specifico articolo della programmazione con dBase V. Lo prepareremo durante le vacanze estive. Per ora descriviamo un utilizzo un po' evoluto dell'ambiente visuale: la realizzazione di una Form per gestire dati provenienti da quattro tabelle.

### Un esempio di utilizzo un po' spinto

Abbiamo voluto provare quanto si possa spingere la complessità di una Form senza arrivare alla programmazione. Abbiamo costruito un'applicazione con quattro Tabelle, legate da tre relazioni, e poi abbiamo provato a realizzare un'unica Form nella quale, in qualche maniera, fossero gestibili i dati provenienti dalle quattro tabelle. La problematica applicativa affrontata è quella classica Clienti, Ordini, Righe e Articoli.

Poiché la Form da realizzare riguarda gli Ordini abbiamo costruito una Query di preparazione dati, partendo dagli Ordini. Abbiamo poi aggiunto la Tabella

Clienti e quella Righe. Essendo partiti dagli Ordini la Tabella dei Clienti (posta in rapporto Uno a Molti con la Ordini) presenta il Cliente che ha eseguito quell'ordine. Dall'altra parte, verso le Righe, la relazione è Uno a Molti, con la parte Uno sostenuta dagli Ordini. In questo caso la situazione si ribalta. Per ogni Ordine, nella Query, appaiono tutte le righe relative. Ciascuna Riga, a sua volta, fa riferimento ad un Articolo e quindi è in grado (essendo la parte Molti della relazione) di mostrare il dettaglio dell'articolo venduto.

Nelle varie figure, dalla 21 alla 23, vediamo le varie fasi dell'operazione. Dapprima abbiamo costruito, e riempito con pochi dati significativi, le Tabelle. Poi abbiamo costruito la Query (fig. 21) stando attenti ad organizzare l'ordine delle Tabelle in modo che la Tabella principale fosse quella degli Ordini. Testato il funzionamento della Query, che deve presentare i dati esattamente nell'ordine voluto (fig. 22), si passa alla costruzione della Form (fig. 23) che può essere realizzata sfruttando il Form Expert e poi ritoccata a mano.

### Conclusioni

Cercheremo di trarre delle conclusioni solo quando, tra un paio di mesi, avremo potuto esplorare a fondo, e quindi presentarvi un po' meglio, l'ambiente di programmazione. Con l'occasione parleremo anche della sottostan-

te tecnologia di accesso ai dati, basata sulle IDAPI (Integrated Database Application Program Interface) che consente di sviluppare applicazioni indipendenti dalla tipologia di Database usata.

Per ora possiamo sicuramente affermare che la massiccia terapia di ringiovanimento, cui gli esperti della Borland hanno sottoposto il dBase, ha dato buoni risultati.

Un utente poco esperto può facilmente maneggiare Tabelle, realizzare Query di medio impegno, costruirsi Form e Report per gestire i propri dati.

Un utente più esperto può spingere, come appena visto, l'acceleratore sulle Query e sulle Form, senza ancora essere costretto non solo a programmare ma neanche ad arrembiare con la Command Window.

Per quanto riguarda la programmazione il dato iniziale, e cioè la compatibilità con le versioni precedenti del dBase, deve tranquillizzare chi utilizza vecchie procedure ed è titubante a passare oltre.

Nel campo della programmazione vera e propria, precisato di nuovo il fatto che aspettiamo il prossimo articolo per fare delle valutazioni, dBase V si scontra con concorrenti agguerriti, sia con il successore del Clipper, l'Object Vision della C.A, che con il Fox Pro 3.0 per Windows della Microsoft. Quest'ultimo prossimamente disponibile in versione 32 bit.

Il confronto sarà interessantissimo. Ne riparleremo.