

Helix Express 3.0

di **Raffaello De Masi**

Finita l'università (avevo ventidue anni, altri tempi, giusto per rimanere in tema!) fui chiamato a servire la patria; manco a dirlo, da buon italiano, pensai di fare di tutto per evitare questo indesiderato impegno, cercando santi patroni e raccomandazioni in mezza regione. Risultato delle mie lagrimose ricerche fu che in due ore mi fecero abile arruolato, per di più nell'arma dei bersaglieri. Cercai a questo punto di limitare il danno, chiedendo ai miei soliti santi patroni di farmi ottenere una destinazione vicino a casa. Un colonnello, che doveva essere la mia carta vincente, mi disse chiaro e tondo

che non poteva certo farmi restare ad Avellino, ma mi avrebbe fatto assegnare a qualche ufficio periferico di Salerno o Napoli. Il risultato fu che la mia destinazione risultò un battaglione operativo a Novara (e meno male che non abbiamo più le colonie in Africa!).

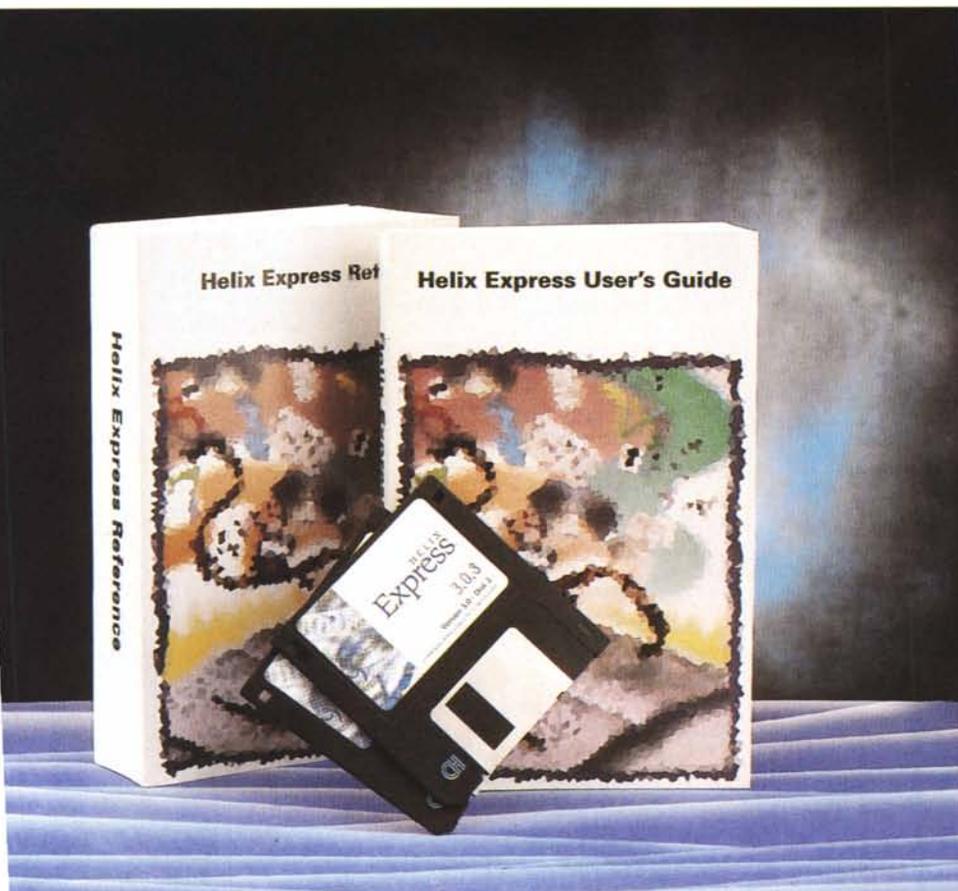
Così caricai armi e bagagli e mi trasferii tra le nebbie del nord; in fondo l'esperienza non fu negativa visto che il mio principale compito svolto durante l'anno di ferma fu quello di fare lezione alla figlia di un maggiore, particolarmente poco votata alla matematica. Sarà stata la santa protettrice dei bersaglieri, sarà stato un minimo di senso

della giustizia celeste che mi compensava del mancato funzionamento delle raccomandazioni avute, la ragazzina cominciò ad ottenere risultati sempre migliori, cosa che si tradusse in licenze sempre più numerose e lunghe.

La ragazzina aveva una notevole collezione di LP che teneva in maniera accurata e aggiornatissima utilizzando una sua speciale tecnica basata su schede di diverso colore in base al genere di musica e all'anno di edizione. Spero che Carlotta, questo il suo nome, mi legga oggi e, se la sua mania dell'ordine le è rimasta, scelga per i suoi archivi Helix Express, il database di cui parliamo oggi.

Helix Express 3, un database sui generis

Occorre che io faccia una piccola precisazione. Chi mi legge da molto tempo sa che fino a un paio di anni fa recensivo su questa rubrica numerosi package disponibili all'estero ma non importati in Italia. A un certo punto (a ben vedere anche a ragione) fui «richiamato all'ordine» dalla direzione della testata che mi indicò di provare, per quanto possibile, solo pacchetti importati e disponibili in Italia, trasformando in eccezione quella che fino ad allora era stata la regola. È questa la prima volta che utilizzo questa chance, vale a dire che presento una prova di un pacchetto non ufficialmente importato in Italia, almeno fino al giugno '95. Il motivo c'è, dato che si tratta, come vedre-



Helix Express 3.0

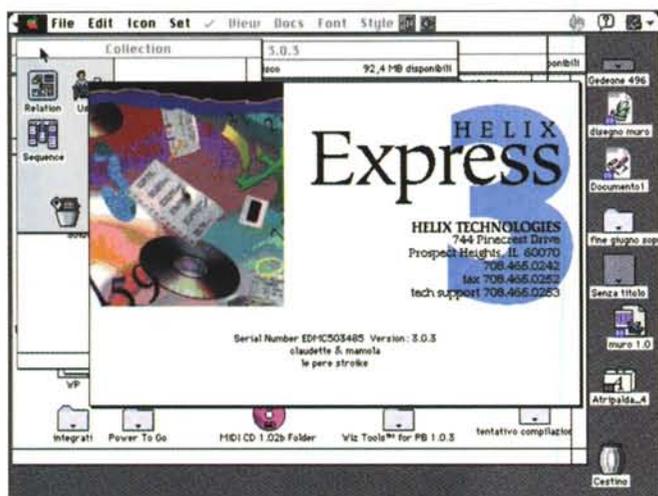
Helix Technologies Inc.
744, PineCrest Dr.
Prospect Hts
IL 60070
USA

Tel. 768-465-0252

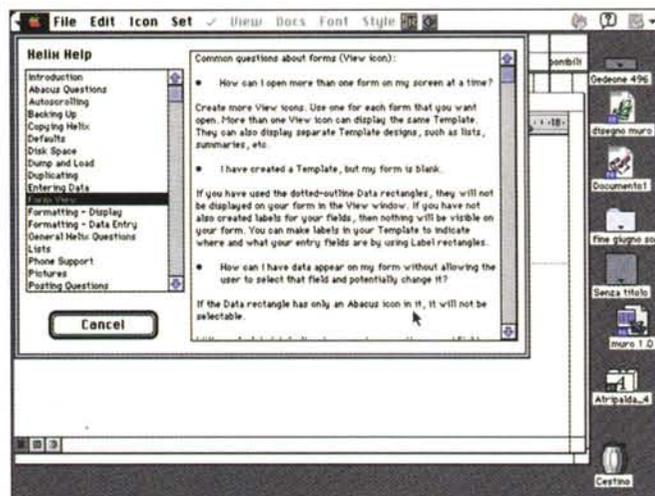
Prezzo in USA:

listino 850.00 \$

road price 650-700 \$



Lo splash screen di Helix Express.



L'help di HE, che, a differenza di quelli comuni, espone, sotto forma di domande e risposta la maggior parte dei problemi che si possono verificare nelle varie sezioni.

mo, di un package dalle caratteristiche eccezionali!

Di HE ho accennato diverse volte su queste pagine, se ben ricorderete, specie quando ho presentato prove di altri database. Si tratta di uno dei package più vecchi esistenti sul mercato: Helix vide la luce nel 1984, assieme alla Odesta, sua casa produttrice. Si trattava di una software house molto dinamica, che produceva anche un integrato e un wp dalle caratteristiche innovative, in un tempo in cui MacWrite era quello che passava il convento e l'unico concorrente era MS Word 1.0, praticamente incontrollabile in quello che faceva.

Helix passò attraverso quattro versioni e divenne, intorno al 1990 Double Helix; Odesta era da un pezzo sparita, e la nuova casa produttrice si chiamava, tout court, Helix Technologies, a significare della grande fiducia e impegno delle nuove forze verso questo pacchetto. Venne poi Express, oggi alla versione 3.0.

Helix Express è, senza preamboli e falsa modestia, uno dei database disponibili in ambiente Mac più potenti, versatili, piacevoli da usare. È un potentissimo DB relazionale, la cui principale caratteristica sta nel fatto che, sebbene sia dotato di un linguaggio di programmazione dei più potenti mai visti in ambiente PC, non chiede all'utente di scrivere neppure una riga di programma, in linguaggi più o meno criptici. Tutto quel di cui si ha bisogno è a portata di mano, sotto forma di «tile», mattonelle

che si trascinano in una finestra bianca e si legano tra di loro a secondo delle necessità.

E questo già avveniva più di dieci anni fa, alla faccia di chi, negli ultimi tempi, si è riempita la bocca di programmazione a oggetti.

Ma procediamo con ordine; il pacchetto è fornito, completo di esempi e di un demo di cui parleremo tra poco, compresso su appena due dischetti HD. Scompresi, questi «liberano» circa 5 mega di software, che comprendono l'applicazione principale, curiosamente (poco più di un megabyte) piccola (almeno per la potenza del pacchetto), i moduli client-server, una serie di utility e un ben nutrito pacchetto di esempi e applicazioni già pronte.

Il programma non è protetto, se si esclude, all'inizio, l'inserimento di una criptica «enablement key» tanto complessa che è ben difficile «centrarla» al primo colpo. Al lancio si entra in un ambiente che più alieno non si può.

Il funzionamento del programma

In ossequio alle più aggiornate regole che impongono, almeno per i database, la norma «Più potente è l'applicazione, tanto maggiore è l'elementarità dei tool a disposizione», anche Helix impone all'utilizzatore di creare, nei minimi particolari, e di customizzare l'ambiente di utilizzo del pacchetto che sta costruendo. Questo vuol dire una posizione diametralmente opposta a quella

di FileMaker, tanto per fare un esempio; in questo si può cominciare a lavorare efficientemente con pochi setup iniziali e un minimo lavoro di finitura dell'ambiente. In Helix, come in 4D e in Omnis, suoi diretti antagonisti, lo shell della base dati va realizzato con cura nei minimi particolari, partendo dalla definizione (ovviamente) dei campi fino a definire i layout di input, output, report, ricerca e così via.

Ma procediamo con ordine. Al lancio di HE, dicevamo, ci si ritrova con una curiosa finestra aperta; obbediente a rigide regole di coerenza e al principio che «ambiente che funziona non si cambia» si spalanca una finestra che, con tutte le altre, ha in comune lo stesso aspetto. Essa è divisa in due parti; quella a sinistra che contiene i tool, quella a destra che contiene le pièce di programma.

Il principio di funzionamento dell'ambiente è piuttosto semplice; al contrario di quel che avviene in altri package, qui si lavora passando dal generale al particolare. In pratica, secondo i più puri canoni della programmazione top-down, si parte dalla struttura generale del pacchetto da realizzare (in HE le strutture di base dati sono chiamate proprio applicazioni) e ogni parte della struttura (record, query, view, report, e così via) viene poi sviluppata, secondo una struttura ad albero più o meno gerarchizzata, «in giù» fino ai particolari più estremi.

Ma facciamo un esempio, in modo

sioni pari a quelle dei campi, pronta ad accogliere i campi, appunto, che saranno indicizzati. Si trascina, così il nome-riga rappresentante un campo nel buco nero, e si stabilisce il primo campo da indicizzare (opzione non sempre disponibile in altri ambienti, l'indice stesso può essere ulteriormente definito in ascendente o discendente). Non esiste limite per la lunghezza dei caratteri da indicizzare.

Finora abbiamo solo lavorato alla costruzione delle fondamenta della base

dati. Ma una base non è solo una semplice raccolta prona di notizie; deve avere la possibilità di manipolarli, modificarli, eseguire calcoli sui suoi valori. Un intero paragrafo, piuttosto corposo, del «Guided Tour» (che al contrario di quanto generalmente si vede in altri pacchetti, è qui un vero e proprio tutorial, abbastanza completo nelle opzioni) ci insegna a creare diverse relazioni, a eseguire calcoli più o meno automatici su di essi, a usare schede con default autocalcolati. Se fino a qui, anche se

con forma autonoma e originale, HE ha seguito canovacci e tecniche comuni ad altri package, è l'occasione buona per discostarsene. È questa la prima volta in cui vediamo in azione i «tile», le mattonelle che rappresentano il linguaggio di programmazione, altamente sofisticato, di Helix.

Che casa è un tile? Semplice, è una keyword, una funzione, un comando che permette di eseguire operazioni più o meno complesse. Si tratta di «oggetti» (e, è opportuno ribadire, questi

Le relazioni (non pericolose) di Helix Express

Tradizionalmente, i dati in un database sono rappresentati come tabelle di dati, legati tra loro da relazioni. Non occorre per questo pensare a complesse teorie, ad esempio una persona e il suo numero di carta di credito sono già una relazione.

Un modo pratico e utile per guardare a un sistema di maneggio dei dati è quello che «ragiona» in termini di come i dati sono organizzati. È fondamentale pensare che l'organizzazione di essi stabilisce una «relazione». Ci sono tre tipi di base di relazioni:

– Uno a uno; è la classica organizzazione dei database piatti; l'esempio precedente è valido. Un altro potrebbe essere la corrispondenza tra nome e patente di guida, o tra figlio e padre (non il contrario, attenzione!).

– Uno a molti; un viaggiatore in un aereo e i suoi voli. Il viaggiatore è unico ma i voli che esso fa sono diversi: è importante considerare, per poter ben capire la successiva relazione, che esiste una sola relazione di ogni persona con un solo aereo in un preciso momento, anche se il viaggiatore ha eseguito diversi viaggi in tempi successivi.

– Molti a molti: alcuni database relazionali, anche di gran nome, non supportano questa relazione. Qui le cose sono un poco più complesse. Si tratta, per restare nell'esempio, delle relazioni che intercorrono tra passeggeri e aeroplani; ogni passeggero ha viaggiato su diversi voli, e ogni aeroplano è stato utilizzato per voli diversi. Per implementare questa sorta di relazione in un database, la struttura sarà normalmente articolata su tre relazioni. In questo caso la relazione Voli contiene informazioni sia riguardo ai passeggeri che agli aerei. È ovvio che una struttura così complessa può essere utilizzata anche per relazioni più semplici; ad esempio, si scoprirà che in un certo volo hanno volato Tizio Caio e Sempronio, mentre in tempi diversi hanno volato sullo stesso aeroplano Gaspere, Melchiorre e Baldassarre (ammesso che quest'ultimo sia riuscito a superare i cani antidroga).

Quando usare un database piatto o relazionale?

Un db piatto è più semplice di uno relazionale. Sebbene molti tecnici e studiosi di base dati non includano nei database propriamente detti questo tipo di programmi, essi hanno avuto, soprattutto nell'area dei micro-computer un notevole successo (si consideri, ad esempio FileMaker). La rubrica telefonica, la guida Monaci, Pirelli, o Michelin, le pagine gialle sono esempi cartacei di db piatti. La difficoltà e la debolezza di questi sistemi è insita nella loro struttura. Un db piatto può infatti manipolare un solo file alla volta; occorre quindi, come minimo, stare attenti a quale base dati aprire per cercare i dati che ci servono.

Un database relazionale può essere invece guardato come una collezione di tabelle separate, i cui valori-campi sono legati tra di loro da relazioni e da procedure integrate che permettono collegamenti diretti anche tra elementi sistemati fisicamente in file diversi. Il vero vantaggio di un database relazionale, come Helix, sta nel fatto che non occorre preoccuparsi troppo di dove i dati sono conservati e di come fare per accedervi. Concettualmente i db relazionali sono «data independent», vale a dire che i dati sono indipendenti dal disegno del database (non a caso quasi tutti i db relazionali conservano separati i blocchi dati dal disegno).

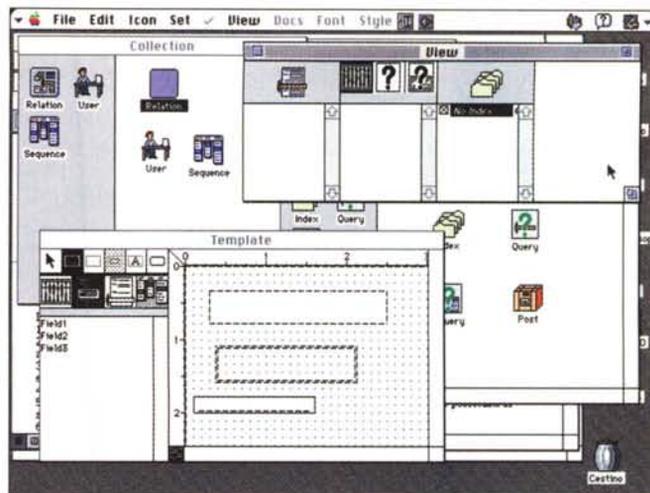
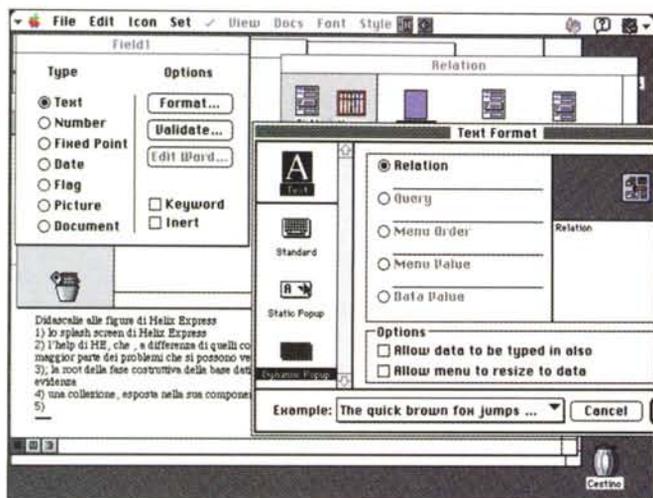
Il vero vantaggio che Helix ha sugli altri è che esso possiede tool più sofisticati per la costruzione del disegno di base. Una Relazione di Helix è analoga a una tabella in un db relazionale più tradizionale; viene assegnato questo diverso nome in quanto in HE una relazione contiene ben più di una tabella, visto che in essa ci sono compresi i tool per la ricerca e l'ordinamento, i formati scheda, i collegamenti con altre relazioni. Ancora, una collezione unica di Helix permette di custodire dati ordinariamente collezionati in molti file. Questa maggiore apparente complessità (come abbiamo già fatto notare, uno dei più famosi db relazionali per Mac non nomina

neppure la relazione «molti a molti») è talmente ben «levigata» in Helix da risultare per nulla fastidiosa, ma addirittura amichevole. Pensate in termini di manipolazione di un ipermercato e delle sue gerarchie-relazioni (prodotti diversi, di diversi reparti, possono essere forniti dallo stesso venditore, o lo stesso tipo di prodotto, ad esempio pneumatici, può essere fornito da diversi fabbricanti) e vedrete che più tool di manipolazione ci sono, già pronti, e più sono i benvenuti.

E allora, di quale tipo di database ho bisogno?

È ovvio che una domanda del genere può essere stata posta solo da un utente avanzato. Per tenere da conto il nostro deposito in banca o la nostra collezione di vini, un db piatto basta e avanza. Per un utente più sofisticato invece il problema cambia.

Costruirsi un database piuttosto ampio e articolato su un motore piatto non è mai complesso; quasi sempre è sufficiente affidare la realizzazione dello stesso anche a un semplice impiegato che conosca un poco il pacchetto e abbia ben presente il problema. Nessuna preoccupazione, in questo caso di mole di dati da manipolare, anche i più piatti dei database maneggiano, oggi, disinvoltamente, migliaia di record senza problemi. Ma, presto o tardi, ci si ritroverà a risolvere i classici problemi connessi con il relazionale; qui i problemi sono di due tipi. Innanzitutto occorre un consulente che conosca bene il pacchetto e sappia costruire un'applicazione efficiente e precisa. Occorre anche prevedere un trasferimento cruento dei dati, in quanto spesso un file a struttura piatta mal si adatta a uno di tipo relazionale (immaginate solamente un cliente le cui generalità si ripetono nel primo caso cento volte e nel secondo una). Certo, si potranno costruire procedure di filtraggio, separazione e smistamento dei dati, ma a costo di quanto lavoro di programmazione, lavoro che sarà utilizzato una volta sola e poi destinato a essere buttato via?



La fase di costruzione di una View, e, in secondo piano, di un Template.

Una fase di definizione di un campo, in questo caso di testo, con, in primo piano, le possibilità di scelta della definizione del campo.

oggetti hanno preceduto, di molti anni, quelli della programmazione OO) che vengono sistemati sul piano della finestra, e «legati» tra loro in vario modo per costruire una procedura o eseguire calcoli successivi o concatenati.

Cerchiamo di capirci con un esempio. Immaginiamo di trovarci nella necessità di dover inserire, nella funzione che stiamo costruendo, un salto condizionale. In un qualsiasi linguaggio avremmo usato una struttura IF ... THEN ... ELSE o simile, eventualmente inserendo negli intervalli, altre strutture condizionali o forzature di uscita. La costruzione di un tale costrutto, specie se molto complesso e articolato, imporrebbe a un programmatore (ad esempio in 4D o Omnis) di costruirsi un'adeguata flowchart o, almeno, uno schema su un foglio. Da qui dovrebbe poi passare alla traduzione nel linguaggio proprio del pacchetto, che, manco a dirlo, è diverso da parrocchia a parrocchia.

Helix ha risolto in maniera brillante il problema; offrendo, anzi imponendo al programmatore la scelta del tool attraverso quella di una piastrella che già contiene il comando; restiamo nell'esempio e vediamo come si fa. Si trascina la mattonella «IF» sulla finestra a destra; essa si configurerà automaticamente con il posto per gli operandi e con le direzioni di uscita.

Lo spazio riservato agli operandi è rappresentato da due «hole», due cavità in cui inserire qualcosa. Verrebbe istintivo batterci dentro un test (ad esempio Campo_1 = TRUE); sbagliato! Nel farlo si trascina un'altra mattonella-test che contiene già il costrutto pre-costruito; e se ci fosse necessità di un AND o un OR; niente paura, aggancia-

mo al nostro test la relativa mattonella e l'operazione di verifica stessa può essere allungata. E le uscite? Esse sono rappresentate da due frecce la cui punta va trascinata nelle mattonelle che eseguiranno le relative operazioni.

Detto così, il procedimento pare fatto apposta per creare uno splendido spaghetti-program, come se ne facevano, una volta, in Basic. Niente di più falso; una procedura scritta in linguaggio HE è, anche se piuttosto complessa, sempre chiara e leggibile da chiunque sappia leggere la particolare sintassi (anche se questo termine è un poco improprio) del linguaggio Helix. Oltre tutto la struttura pittorica utilizzata aiuta molto, permettendo di costruire, in pratica flowchart senza il fastidio di prepararle prima sulla carta.

Bene, abbiamo capito come costruire procedure. E per i calcoli? Nessun problema, si agisce allo stesso modo. Dobbiamo eseguire una addizione? Trasciniamo la relativa mattonella sul piano laterale e riempiamo i buchi degli addendi con quel che si desidera (campi, funzioni predefinite, semplici valori). Questo valore ci serve in un altro calcolo? Definiamo prima la mattonella di questo e poi trasciniamo la freccia del risultato nel buco dell'operatore relativo.

Subform, posting e menu

Per le prime poco da dire; si tratta di uno dei mattoni di base dei database strutturati. Il posting è invece più curioso e rappresenta una caratteristica piuttosto originale di HE.

Come acquistare in USA

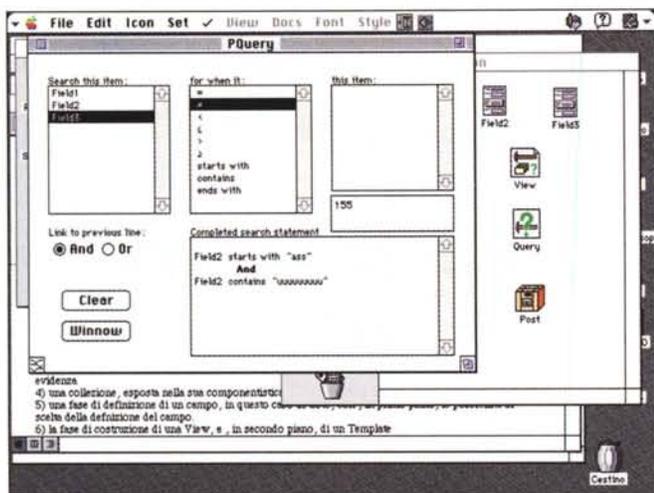
Ripeto qui alcune notizie, già altre volte pubblicate in questa rubrica. La tecnica più sicura è quella della carta di credito (quasi universalmente accettata VISA e MasterCard, meno AMEX); basta comunicare il nostro numero (anche per fax) e si è immediatamente accreditati per la spedizione (che, per corriere, impiegherà non più di 2-3 giorni).

Per motivi di sicurezza, la merce verrà spedita solo all'indirizzo del titolare della carta di credito.

Questo per evitare che «ignoti» rubino merce con questo sistema, indicando ad esempio un indirizzo d'appoggio (Mario Rossi, presso Copisteria «Pizza e Fichi», ecc., ecc.), addebitando il costo all'ignaro malcapitato di turno.

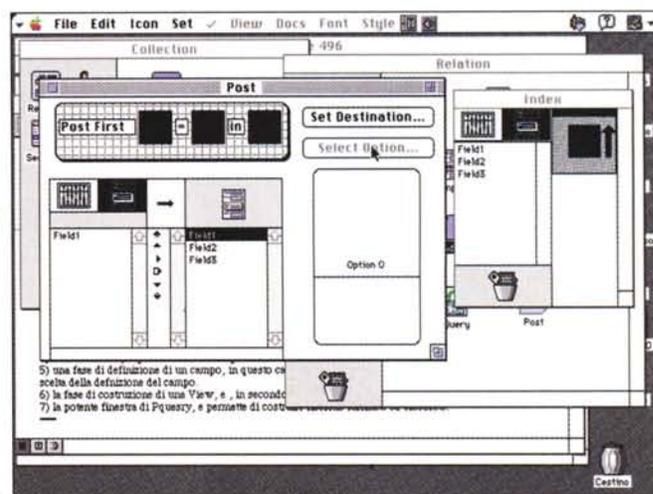
Un altro sistema è quello di spedire direttamente un assegno personale (ormai, da diverso tempo, quasi tutte le banche ammettono l'uso dei loro assegni anche per valute estere), o un vaglia postale (lentissimo e non privo di problemi).

All'arrivo in Italia si verrà chiamati direttamente dall'importatore che chiederà l'autorizzazione allo sdoganamento; dopo una giornata circa la merce è a casa, a fronte del pagamento dell'IVA (19%) e dei dazi (mediamente dal 4 al 5.5% a seconda della bontà d'animo del doganiere). Se siete fortunati, in tre-cinque giorni la merce è arrivata.



La potente finestra di Pquery, che permette di costruire ricerche raffinate ed elaborate.

La finestra di Post e, a fianco, quella di costruzione degli indici. Si noti, nella prima, l'uso dei «tile», così diffuso nel pacchetto.



Per bene intendere cosa sia il posting (che letteralmente significa impostare, nel senso di mettere in una casella della posta, ma anche consegnare in gran fretta) bisogna tenere bene a mente che le relazioni (che, per grandi linee, possono essere considerate come i file di altri db relazionali) possono essere tra di loro combinate anche su lunghe gerarchie. Il posting consente di creare relazioni per definire e maneggiare valori parziali, prelevare continuamente informazioni da una relazione e passarli a un'altra, aprire automaticamente una form basata su determinati criteri.

Chiaro, vero? No? In altri termini il Post è direttamente collegato con un trigger, un grilletto che scatta appena o ogni volta si verifica un certo evento. Immaginate l'utilità di un'opzione del genere, per esempio, in una gestione di magazzino, quando si desidera monitorare giacenze delle scorte o clienti che ritardano nei pagamenti.

E adesso parliamo di un'altra cosetta interessante. Ricordate all'inizio che vi ho detto che la prima finestra, quella cosiddetta «main», aveva tre icone? La prima permetteva, abbiamo visto, la definizione dei campi, delle form, la seconda impostava l'ambiente «user», la terza costruiva sequenze.

Una delle maggiori responsabilità di un database è quella di ordinare e cercare dati in base alle specifiche dell'utente, nella maniera più efficiente possibile. Non a caso l'efficienza di un db si misura eseguendo benchmark costruiti su disegni particolarmente complessi. Coerentemente, ancora una volta, con le più attuali e avanzate linee di tendenza, HE offre, anche per un'operazione che potrebbe sembrare piuttosto

univoca, numerosi interessanti opzioni. Si va, infatti, da QuickQuery cosiddette al volo (ricerche rapide e semplici, del tipo «Portami alla scheda di Marco Marinacci») a FormQuery, classiche ricerche eseguite su una form già definita, ad Abacus Query, che incorporano anche calcoli automatici su cui la stessa ricerca si basa, a PowerQuery. Si tratta, quest'ultima, di un'opzione introdotta in questa versione 3 che sana quello che era il tallone d'Achille delle precedenti release. Questo tool molto potente e flessibile, permette ricerche complesse basate su molteplici criteri di ricerca, anche su campi multipli e

Abachi (campi calcolati); in teoria sarebbe possibile definire ricerche basate anche su What... If, vale a dire, ad esempio: «Cerca tutti i clienti che, tra quindici giorni, raggiungono più di tre mesi di ritardo nei pagamenti e che non abbiano ancora versato acconti attraverso bonifico sul conto corrente XXXX, escludendo quelli che hanno fatto bonifici sul conto YYYY».

Una delle cose più interessanti nella realizzazione di un pacchetto applicativo, sia costruito con il nostro Helix che con qualunque generatore di applicativi, è la possibilità di costruirsi menu personali. Occorre riconoscere che in questo campo i database hanno offerto, specie ultimamente, mezzi veloci, facili e pratici per la bisogna; quasi dappertutto si opera allo stesso modo: costruzione di una procedura, definizione con un nome, assegnazione a un menu e a una voce relativa.

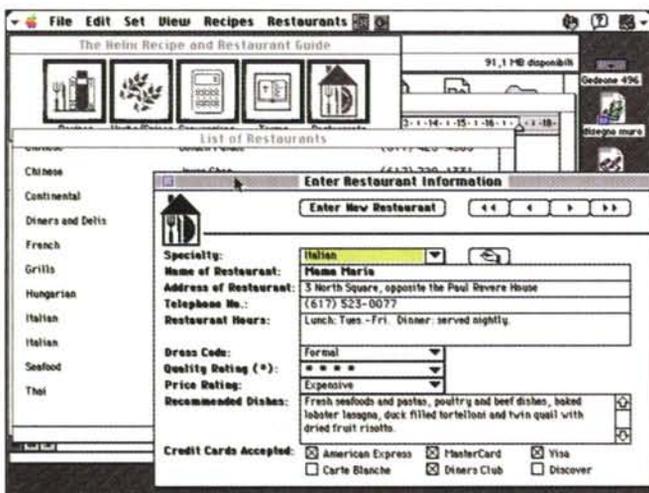
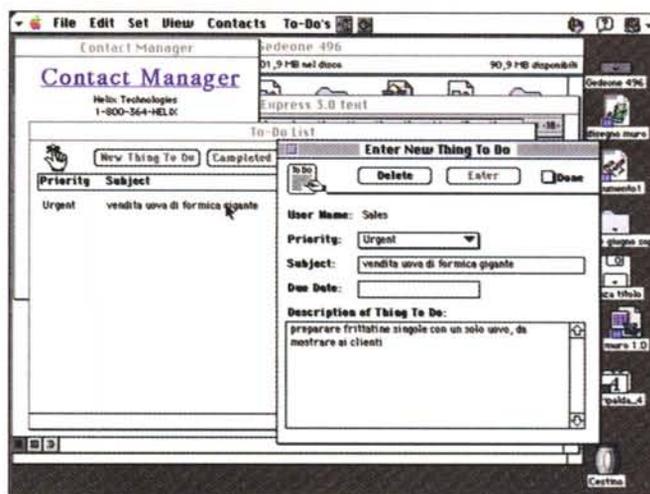
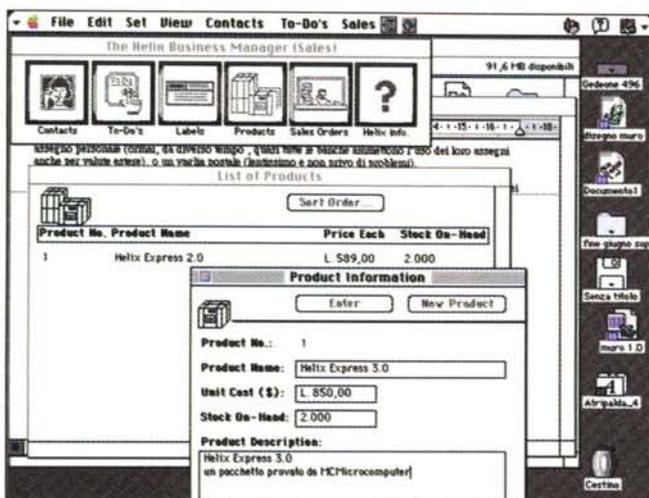
Anche qui la struttura del procedimento è la stessa; viene costruita una sequenza (sinonimo in HE di Procedura, Script), sempre, non dimentichiamolo, attraverso la realizzazione del solito «pavimento» di mattonelle, e la si combina con una voce di menu customizzata (i menu possono essere creati addirittura automaticamente). Poiché una sequenza può a sua volta chiamare un menu è evidente come si possano realizzare strutture sofisticate in breve tempo; il rovescio della medaglia, comune comunque a tutti questi ambienti, è la facilità con cui si costruisce «garbage», ciarpame di cui si dimentica la funzione e che si trascina appresso senza alcuna necessità. Poiché HE invoglia molto ed è, in questo ambiente, molto familiare e facile, occorre avere un poco di autodisciplina nel gestire

PRO

- Data base dell'ultima generazione, relazionale, potente e flessibile
- Agevole da programmare, in quanto non richiede alcuna conoscenza di linguaggi di programmazione
- Ambiente client-server del tutto trasparente
- Numerose utility preziosissime tra cui un custom helper

CONTRO

- L'interfaccia a oggetti richiede un certo periodo di assuefazione
- Manca un qualsiasi compilatore; le runtime sono abbastanza costose.



Tre videate ricavate dagli esempi forniti col pacchetto. Tra l'altro, molto interessante, è presente una base dati di sviluppatori Helix, suddivisi per categorie merceologiche.

senza problemi, scritto, immagini, documenti esterni; mentre Recipe and Restaurant è una guida Michelin elettronica che arriva fino a indicare gli ingredienti dei vari piatti.

E, giusto per chiudere, Contact Manager e Business Manager sono potenti agende, private e commerciali, per gestire, ad esempio, un pacchetto clienti o i nostri contatti privati.

Conclusioni

Di Helix Express diverse persone mi avevano chiesto notizie, avendolo io in più riprese nominato sulle pagine della rivista, probabilmente in occasione di prove di altri database.

Oggi ho l'occasione di mostrare come è strutturato e di quali sofisticazioni e prestazioni è capace. Non è velocissimo come FoxPro né come Panorama (un vero fulmine quest'ultimo, che soffre del fatto di non essere relazionale; ma entro l'anno dovrebbe uscirne la versione 3, e chissà cosa ci offrirà), ma si impara a usare e programmare in un pomeriggio; dopo di che è assolutamente divertente nell'uso, non fosse altro che anche le più complesse procedure non richiedono la scrittura neppure di un rigo di codice. Dotato di un manuale chiarissimo ed estremamente illustrato (anche se il materiale cartaceo supera le mille pagine), fornito di alcune applicazioni già pronte ben realizzate, gira su macchine anche piccolissime, pur essendo ottimizzato per i PowerPC.

È effettivamente un'efficace alternativa ad altre soluzioni per chi non desidera impelagarsi in listati criptici e complessi disegni di strutture.

MC

questa area del pacchetto, a scampo di forti mal di testa quando si farà l'inevitabile debug dell'applicazione.

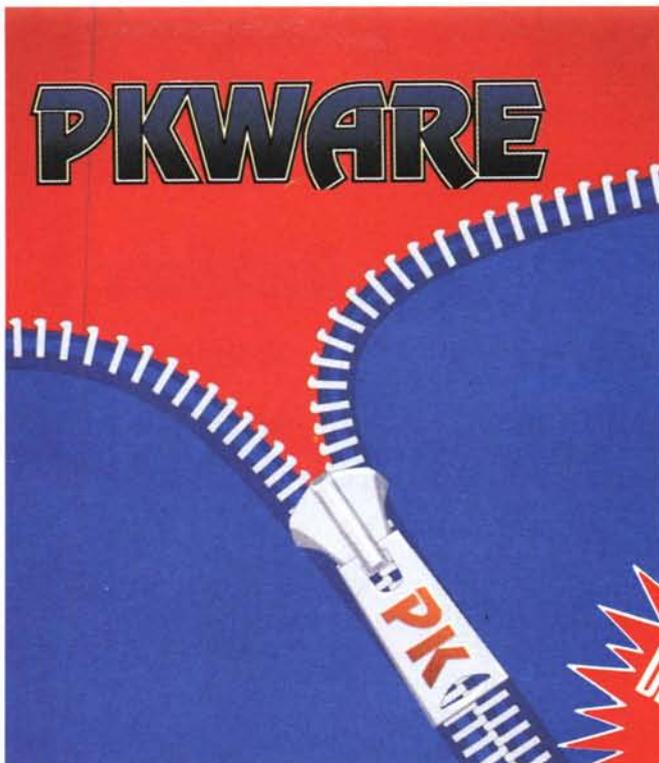
L'ambiente multiutenza e le applicazioni già pronte fornite

Come già accennavamo all'inizio, il pacchetto di base offre un completo ambiente Client_Server perfettamente funzionante, tranne che per essere limitato, nei tempi di funzionamento, a un'ora. Si tratta di un pacchetto di «assaggio» delle funzionalità di questa architettura, che permette, mettendo (come consigliato nel manuale) accanto almeno tre Macintosh, di sperimentare sul campo la praticità e la facilità d'uso insita in questa implementazione.

Qui HE si comporta, giustamente, da pacchetto ad alto livello, evitando

all'implementatore-gestore le seccature insite nella gestione dei semafori e della gerarchia dei tempi d'accesso. Lavorare su un network o su una singola stazione è praticamente indifferente e trasparente.

E infine, cosa quasi mai vista altrove, HE offre un completo tutorial degli esempi offerti come librerie. Si tratta di applicazioni ben diverse da quelle che si vedono altrove, e che non sono altro che modesti esempi di applicazioni, alla fin fine di scarso interesse ed utilizzabilità. Quelle presenti in Helix sono applicazioni eleganti e sofisticate, utilizzabili tal quali per un impiego professionale, o apribili per essere customizzate o visualizzate per imparare le più segrete e sofisticate tecniche di programmazione. Document Manager, ad esempio, è una raffinata base dati che manipola



Licenza - di zippare

Quando Phil Katz fondò la PKWare, nessuno avrebbe potuto immaginare che il suo **PKZip** sarebbe in breve diventato lo standard di mercato tra i programmi di compressione. Oggi gli utenti di **PKZip** si contano a milioni in tutto il mondo, tanto che i termini "zippare" e "unzippare" sono entrati nel linguaggio comune.

Definire **PKZip** un programma di compressione - anche se il migliore e più diffuso, secondo il mensile **BYTE** - è però riduttivo: l'opzione "multivolume", per esempio, ne fa un ottimo strumento per il back-up.

Naturalmente per usare legalmente **PKZip** ci vuole una licenza, che solo noi in Italia siamo autorizzati a concedere. La licenza vi dà il diritto di ottenere la versione corrente e ufficiale del programma, il manuale in Italiano, assistenza e supporto telefonico.

E da oggi c'è un motivo in più per non aspettare: fino al rilascio di

PKZip per Windows - quello originale - regaliamo a tutti Drag&Zip, un programma per la creazione e gestione dei file ZIP sotto Windows.

Non meno importanti e note di **PKZip** sono le altre utility di compressione targate PKWare:

- **PKLite**, che serve a comprimere i file eseguibili in modo "trasparente", lasciandoli cioè eseguibili, ma

riducendone l'occupazione su disco mediamente del 45%.

- **Data Compression Library**, che vi permette di incorporare la tecnologia di compressione nelle vostre applicazioni DOS e Windows.

Per avere "licenza di zippare" compilate **oggi stesso** il tagliando e rispettelo per posta al nostro indirizzo, oppure per fax al numero **02/65.55.061**

SI inviatemi la versione registrata dei prodotti contrassegnati, con licenza d'uso su singolo PC. I prezzi comprendono IVA e spese di spedizione postale:

Pacchetto **PKZip** (programmi PKZIP, PKUNZIP, PKSFx) con **Drag&Zip in regalo** L. 138.000

PKLite standard (manuale in Inglese) L. 98.000

PAGAMENTO versamento c/c postale 25812207 (allego ricevuta)

allego assegno vaglia postale (allego ricevuta)

contrassegno postale (aggiungere L. 8.000 al totale)

DESIDERO MAGGIORI INFORMAZIONI SU Data Compression Library

Licenze multiple/distribuzione Come diventare rivenditore

NOME/COGNOME

INDIRIZZO

CAP CITTA'

TEL FAX



Ultimobyte
 Editrice S.r.l.
 Via A. Zeron, 10
 20124 Milano
 Tel. 02/66.98.31.33
 Fax 02/66.98.32.33