

Verso l'SQL Sintassi relazionale

di Francesco Petroni e Claudio Petroni

Nel numero scorso abbiamo iniziato uno «slalom parallelo» tra dBASE IV e l'SQL, in cui sulla pista di sinistra appaiono comandi relazionali tipici del DOT PROMPT del dBASE e su quella di destra la versione SQL. Obiettivo è quello di illustrare ai molti che conoscono il dBASE, il linguaggio strutturato di Query (SQL, appunto), cui tutti i produttori fanno un po' la corte, perché viene ritenuto il linguaggio che permetterà l'unificazione

L'SQL nasce come applicazione pratica delle teorie relazionali di CODD, e si è diffuso dapprima nel mondo Mainframe IBM e poi via via «in giù» verso il PC.

Esistono altri linguaggi che seguono tali teorie, ma nascendo in altre famiglie, hanno altri nomi ed altri comandi. Citiamo il QUEL e il QBE, di cui esistono varie versioni per diverse categorie di macchine e realizzate da diversi costruttori.

L'ultimo, il Query by Example, la cui operatività si basa non su comandi ma su simbologie da inserire in «scheletri» che rappresentano le strutture dei vari archivi, ha anche una ottima versione su PC, con il Paradox della Borland.

La nostra Base Dati

Nel precedente articolo, abbiamo gettato le basi per cimentarci nella interrogazione di un sistema di archivi più complesso.

In figura 1 vediamo quello che utilizzeremo per le nostre esercitazioni e del quale è opportuno illustrare brevemente la struttura.

Gli archivi sono 5 e sono stati studiati per rappresentare, nella maniera più semplificata ed immediata possibile, un sistema per la gestione di Ordini di Acquisto.

Proviamo ad immaginare il mondo di un fornitore di materiali di consumo per uffici: l'universo in cui si muove è essenzialmente formato da Articoli da vendere, Clienti che comprano e Vendi-

tori che effettuano la vendita per suo conto.

È facile per ora pensare ad una terna di archivi che contengano i dati essenziali concernenti Articoli, Clienti e Venditori:

- l'archivio Clienti conterrà un codice cliente, il nome del cliente, etc...
- l'archivio Venditori sarà formato dal codice venditore, dal nome venditore, etc...
- l'archivio Articoli sarà formato dal codice articolo, dalla descrizione articolo, dal prezzo unitario, etc...

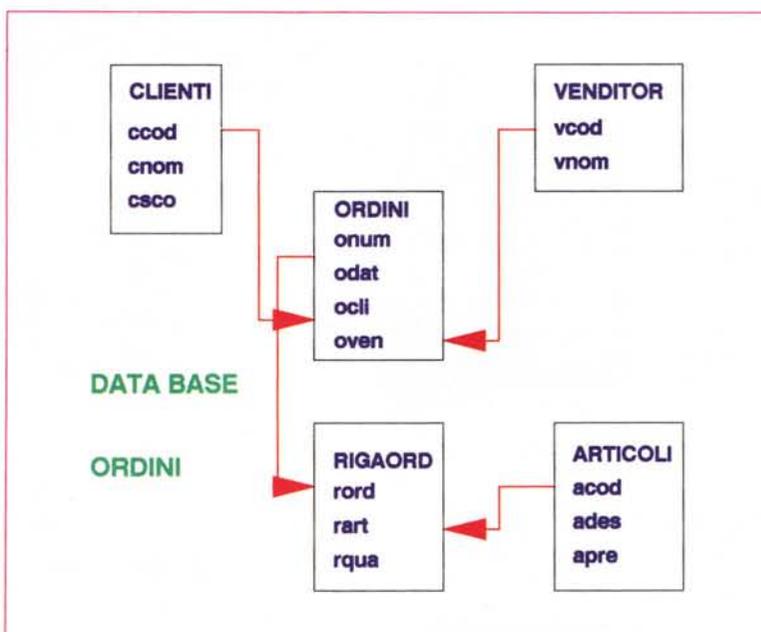


Figura 1 - Schematizzazione del Data Base utilizzato per gli esercizi. Nei vari articoli, sia prove che rubriche, che scriviamo sui prodotti che trattano archivi relazionali, cerchiamo di utilizzare sempre lo stesso caso studio, basato su cinque archivi e quattro relazioni. Rappresenta quindi un test abbastanza significativo, ma sufficientemente leggero da poter essere seguito dal lettore interessato.

Non occorre certo ripetere che un archivio può essere rappresentato graficamente come una tabella in cui sono raccolte informazioni organizzate in righe e colonne, dove per ogni riga, per esempio nell'archivio Clienti, sono registrate le informazioni relative ad un solo cliente e dove per ogni colonna abbiamo la registrazione di uno stesso elemento informativo, per esempio il nome. In questa tabella la chiave per individuare un cliente potrebbe essere il codice cliente.

Fatalmente, al momento del verificarsi dell'evento «Ordine di Acquisto», gli elementi che costituiscono il mondo del fornitore, entrano in contatto:

— un cliente telefona al suo fornitore e chiede di parlare con un venditore perché la sua ordinazione possa essere annotata;

— comunica la sua intenzione di comprare e il venditore prende il blocco dove registra gli ordini annotando, accanto al numero dell'ordine, data dell'ordine ed il cliente;

— a questo punto il cliente elenca gli articoli che intende comprare mentre il venditore annota, uno per riga, il codice dell'articolo e la quantità ordinata. Alla fine della telefonata si salutano cordialmente.

In questa operazione compaiono tutti gli elementi che poco fa abbiamo indicato come costituenti il mondo del nostro fornitore.

Ma come potremmo organizzare in forma tabellare le informazioni essenziali utili alla ricostruzione di questo evento? Procediamo per gradi.

Le prime informazioni riguardano la collocazione temporale dell'evento e sono il numero dell'ordine e la data dell'ordine. A queste vanno aggiunte le informazioni relative al cliente e al venditore, che sono i protagonisti dell'evento.

Potremmo quindi immaginare una tabella ordini dove la riga raccoglie le informazioni che abbiamo appena elencato e dove l'elemento di identificazione (la chiave) è il numero d'ordine.

Ma quante delle informazioni relative al cliente e al venditore devono trovare posto in questo archivio? Certamente non tutte quelle a disposizione (quelle che sono già registrate negli archivi Clienti e Venditori), ma solo i codici cliente e venditore. Attraverso questi codici sarà in seguito possibile rintracciare le informazioni collegate.

Tornando ora all'evento «Ordine d'Acquisto», le informazioni che ancora non

sono state registrate sono quelle relative al dettaglio dell'ordine.

Ipotizzando che il nostro cliente abbia ordinato solo un articolo, potremmo pensare di aggiungere alla riga della nostra tabella ordini le informazioni relative all'articolo ordinato, la quantità ed il costo.

Ma questa soluzione è sbagliata e si presta a due generi diversi di osservazioni:

— prima osservazione: per ragioni analoghe a quella per cui abbiamo inserito nella tabella solo i codici cliente e venditore, sarà sufficiente specificare solo il codice articolo e la quantità (rintracciando, attraverso il codice articolo, nell'ar-

chivio articoli, quello ordinato, avremo a disposizione anche descrizione e prezzo unitario).

— Seconda osservazione: se il nostro cliente avesse ordinato più di un articolo, dove avremmo registrato i dati della seconda «Riga di Ordine»? Se lo avessimo fatto nell'archivio ordini, avremmo necessariamente dovuto impiegare una nuova riga, ripetendo anche le prime informazioni riguardanti la collocazione temporale dell'evento (numero ordine, data ordine, codice cliente e codice venditore).

La soluzione migliore è quella di creare un archivio «Righe di Ordine» che contenga le righe di dettaglio dell'ordine

```

fig.A RELAZIONE SEMPLICE

SELE 1
USE ORDINI
SELE 2
USE CLIENTI INDEX CL_COD
SELE 1
SET RELA TO OCLI INTO B
LIST ONUM,ODAT,B->CCOD,B->CNOM;
FOR ODAT=CTOD("03/01/89")

SELECT ONUM,ODAT,CCOD,CNOM
FROM ORDINI,CLIENTI
WHERE OCLI=CCOD
AND ODAT=CTOD("03/01/89");

fig.B RELAZIONE DOPPIA

SELE 1
USE ORDINI
USE CLIENTI INDEX CL_COD IN 2
USE VENDOR INDEX VE_COD IN 3
SET RELA TO OCLI INTO B,OVEN INTO C
LIST ONUM,ODAT,OCLI,B->CNOM,;
OVEN,C->VNOM;
FOR ODAT=CTOD("03/01/89")

SELECT ONUM,ODAT,OCLI,CNOM,OVEN,VNOM
FROM ORDINI,CLIENTI,VENDOR
WHERE OCLI=CCOD AND OVEN=VCOD
AND ODAT=CTOD("03/01/89");

fig.C RELAZIONE PASSANTE

SELE 1
USE ORDINI
USE RIGAORD INDEX RI_NORD IN 2
USE ARTICOLI INDEX AR_COD IN 3
SELE 2
SET RELA TO RART INTO C
SELE 1
SET RELA TO ONUM INTO B
SET SKIP TO B
LIST ONUM,B->RORD,B->RART,C->ADES

SELECT ONUM,RORD,RART,ADES
FROM ORDINI,RIGAORD,ARTICOLI
WHERE ONUM=RORD AND RART=ACOD;

fig.D ELABORAZIONE LATO ARCHIVIO ORDINI - SOLUZIONE INTERMEDIA

USE ORDINI INDEX OR_NUM IN 2
USE CLIENTI INDEX CL_COD IN 3
USE VENDOR INDEX VE_COD IN 4
SELE 2
SET RELA TO OCLI INTO C,OVEN INTO D
LIST OFF ONUM,ODAT,OCLI,C->CNOM,;
C->CSCO,OVEN,D->VNOM

SELECT ONUM,ODAT,OCLI,CNOM,CSCO,OVEN
VNOM
FROM ORDINI,CLIENTI,VENDOR
WHERE OCLI=CCOD AND OVEN=VCOD;

```

Figura 2 - Comandi per generare relazioni. Ci appoggiamo, anche in questo secondo articolo, al dBASE IV, che contiene un modulo SQL semplificato, ma che ci permette di lavorare con due linguaggi e un unico ambiente. Per quanto riguarda i comandi relazionali del dBASE IV, questi sono stati potenziati rispetto a quelli presenti nel dBASE III.

```

fig.E  SOLUZIONE FINALE LATO RIGHE

USE ORDINI INDEX OR_NUM IN 2      SELECT RORD,RQUA,APRE,RQUA*APRE,OCLI
USE CLIENTI INDEX CL_COD IN 3      OVEN
USE VENDOR INDEX VE_COD IN 4      FROM RIGAORD,ARTICOLI,ORDINI
USE ARTICOLI INDEX AR_COD IN 5    WHERE RORD=ONUM AND RART=ACOD;
SELE 2
SET RELA TO OCLI INTO C,OVEN INTO D
SELE 1
USE RIGAORD INDEX RI_NORD
SET RELA TO RORD INTO B,RART INTO E
LIST RORD,RQUA,E->APRE,RQUA*E->APRE,;
B->OCLI,B->OVEN

fig.F  UTILIZZO DI TUTTI GLI ARCHIVI

USE ORDINI INDEX OR_NUM IN 2      SELECT RORD,RART,ADES,APRE,RQUA,OCLI
USE CLIENTI INDEX CL_COD IN 3      CNOM,CSCO,OVEN,VNOM,
USE VENDOR INDEX VE_COD IN 4      STR((RQUA*APRE*CSCO)/100,6)
USE ARTICOLI INDEX AR_COD IN 5    FROM RIGAORD,ARTICOLI,ORDINI
SELE 2                             CLIENTI,VENDOR
SET RELA TO OCLI INTO C,OVEN INTO D WHERE RORD=ONUM AND RART=ACOD AND
SELE 1                             OCLI=CCOD AND OVEN=VCOD;
USE RIGAORD INDEX RI_NORD
SET RELA TO RORD INTO B,RART INTO E
LIST OFF RORD,RART,E->ADES,;
E->APRE,RQUA,B->OCLI,C->CNOM,;
C->CSCO,B->OVEN,D->VNOM,;
STR((RQUA*E->APRE*C->CSCO)/100,6)

fig.G  ELABORAZIONE LATO ARCHIVI ANAGRAFICI - PRIMA FASE

SELE 3                             SELECT CCOD,CNOM,ONUM,ODAT,OVEN,VNOM
USE VENDOR INDEX VE_COD           FROM CLIENTI,ORDINI,VENDOR
SELE 2                             WHERE CCOD=OCLI AND OVEN=VCOD;
USE ORDINI INDEX OR_CODCLI
SET RELA TO OVEN INTO VENDOR
SELE 1
USE CLIENTI INDEX CL_COD
SET RELATION TO CCOD INTO ORDINI
LIST CCOD,CNOM,B->ONUM
SET SKIP TO B
LIST CCOD,CNOM,B->ONUM,B->ODAT,;
B->OVEN,C->VNOM

```

Figura 3 - Relazioni multiple lato ordini o righe. Chiarito il problema, e la sua schematizzazione può essere considerata la formalizzazione del fatto che il problema è stato risolto, occorre semplicemente aprire gli archivi e le relazioni ed eseguire le interrogazioni. La realtà del Data Base può essere «aggregata» da varie parti. Si sceglierà come archivio principale quello che permette di ottenere elaborazioni da un voluto punto di vista.

identificate dal numero dell'ordine di appartenenza: numero ordine, codice articolo e quantità ordinata.

Siamo arrivati quindi a stabilire che per gestire, ottimizzandoli, i dati relativi all'evento «Ordini d'Acquisti» sono necessari almeno 5 archivi. Il numero sale se le necessità di gestione aumentano di complessità, (fatturazione, magazzino, fornitori del fornitore)...

Le relazioni sottostanti l'applicazione

Abbiamo già parlato di relazioni, ma adesso occorre applicare le cose dette alla nostra nuova situazione.

Immaginiamo quindi i nostri cinque archivi e proviamo a «collegarli» tra loro (vedi fig. 1). Che cosa intendiamo innanz-

zitutto con «collegare»: la possibilità, attraverso una informazione contenuta in un archivio, di ottenerne altre in altri archivi.

Per esempio, leggendo una riga dell'archivio Ordini veniamo a sapere che l'ordine 100 è stato fatto in data 1/1/90 dal Cliente A012 al Venditore Z001. Tenendo conto del fatto che esistono altri quattro archivi, mi è possibile ricostruire l'evento in maniera più leggibile? Certamente sì, mettendo in relazione i codici appena elencati con gli altri archivi a cui appartengono.

Volendo entrare ancora un po' in più in dettaglio, possiamo dire che i collegamenti tra archivi (le relazioni) possono essere classificati: ricordate le definizioni Uno a Molti, Uno a Uno, Molti a Molti? Bene, le relazioni si classificano

in questo modo. Proviamo ora ad applicare queste classificazioni ai nostri archivi:

— Clienti/Ordini: la relazione è una relazione Uno a Molti perché per un cliente esistono molti ordini;

— Ordini/Clienti: la relazione è una relazione Uno ad Uno perché per un ordine esiste un cliente. Quindi una relazione Uno a Molti, vista dal lato Molti, in pratica diventa Uno a Uno;

— Venditori/Ordini e Ordini/Venditori: queste relazioni sono identiche alle due precedenti;

— Ordini/Riga Ordine: relazione Uno a Molti;

— Riga Ordine/Ordine: relazione Uno a Uno;

— Articolo/Riga Ordine: relazione Uno a Molti;

— Riga Ordine/Articolo: relazione Uno a Uno.

Ragionando su queste relazioni (ribadiamo che è molto importante, al di là delle varie definizioni, intuire..) ci accorgiamo che là dove la relazione è Uno a Molti è stato utile suddividere le informazioni su più archivi. Infatti, in questi casi, il non farlo avrebbe comportato gravi problemi di ridondanza, manutenzione e correttezza dati.

Ma che relazione c'è, per esempio, tra Clienti e Venditori? La relazione tra questi due archivi è di tipo Molti a Molti (ogni cliente ha rapporti con più venditori e ogni venditore ha rapporti con più clienti), di un tipo quindi che determina l'esigenza di un archivio intermedio, in relazione 1 a 1 con i clienti e i venditori, matrice tra i due.

Una corretta individuazione delle relazioni ci permette di «percorrere» la nostra base di dati in tutte le direzioni: possiamo infatti, partendo da un cliente risalire a tutti i suoi ordini, ai venditori con cui è entrato in contatto fino a rintracciare la descrizione di tutti gli articoli da lui ordinati.

Analogamente sarà possibile, partendo da un articolo, avere un elenco delle righe ordini in cui compare, e conseguentemente un elenco degli ordini, con corredo di data vendita, cliente e venditore.

In questi casi sfruttiamo la possibilità di passare da un'informazione all'altra percorrendo il reticolo di relazioni tra gli archivi (se A è in relazione con B e B è in relazione con C, possiamo lavorare su A accedendo alle informazioni di C attraverso B).

Quest'ultimo periodo ci suggerisce che forse è giunto il momento di passare a degli esempi concreti a dimostrazione ed esemplificazione di quanto fin qui detto.

Analisi degli esempi proposti

Continuando con lo stesso metodo adottato nell'articolo precedente, metteremo a confronto comandi dBASE e comandi SQL (si fa riferimento all'SQL che gira sotto dBASE, meno sofisticato di altri SQL, ma più adatto ai nostri limitati scopi dimostrativi).

Gli esempi riguardano appunto le relazioni sottostanti i nostri 5 archivi e, cominciando da situazioni molto semplici, vedremo tutti i tipi di relazioni e il loro funzionamento in situazioni differenti.

Figura A - La relazione tra gli archivi Ordini e Clienti è di tipo 1 a N, visto che per ognuno dei clienti troveremo più ordini. Come potete vedere, le istruzioni utilizzate in dBASE sono sostanzialmente quelle necessarie all'apertura archivi e al lancio della relazione. Una volta terminata questa fase preliminare, la disposizione degli archivi e le caratteristiche delle relazioni possono essere modificate solo da uno specifico comando dBASE, in assenza del quale la situazione resta immutata e tutti i valori degli archivi relazionati possono essere contemporaneamente referenziati.

In SQL è tutta la situazione che va, anche se in maniera meno dettagliata e con meno attenzione alle varie specifiche relative agli archivi, ogni volta riproposta. Questo è solo apparentemente uno sforzo, in quanto anche dando un'occhiata superficiale agli esempi, appare evidente che comunque l'impegno è minore.

Prima ci continuare, vorremmo stabilire, per semplicità, delle convenzioni che utilizzeremo da questo momento in poi: ci riferiremo all'archivio aperto nella Select, che alla fine delle varie aperture rimane attiva, come all'archivio di lavoro, mentre gli archivi in relazione con quello di lavoro potranno essere indicati come archivi di decodifica o archivi di ricerca, a seconda che la relazione sia 1 a 1 o 1 a N.

Ritornando quindi all'esempio, possiamo dire che l'archivio di lavoro è l'archivio Ordini, mentre l'archivio di decodifica è l'archivio Clienti. Questo infatti viene utilizzato per decodificare il codice cliente presente nell'archivio Ordini sulla base di una relazione 1 a 1 (un ordine un cliente), relazione che permette, ovviamente, di ricavare anche le altre informazioni relative al cliente. In SQL è sufficiente elencare i campi e gli archivi interessati senza specificare altro se non i campi la cui uguaglianza determinerà poi il criterio di relazione.

Figura B - La relazione può essere lanciata anche per un numero di archivi

maggiore di 2. Nell'esempio a cui ci riferiamo troviamo infatti tre archivi: l'archivio di lavoro è sempre l'archivio Ordini, questa volta messo in relazione con due archivi di decodifica. La relazione attivata è, in entrambi i casi, di tipo 1 a 1, poiché l'archivio Ordini ha con Clienti e Venditori lo stesso rapporto (un ordine è relativo ad un cliente ed un venditore). La tabella risultante da un'interrogazione di questo tipo avrà (come, del resto, per l'esempio precedente) lo stesso numero di riga dell'archivio Ordini.

Figura C - Il terzo esempio ci fa vedere, come già accennato, che, dato un archivio A in relazione con un archivio B, per mezzo di una seconda relazione, questa volta tra B ed un terzo archivio C, possiamo risalire dalle informazioni in A a quelle in C.

Ordini sarà quindi in relazione con Rigaord e Rigaord sarà in relazione con articoli. Queste due relazioni costituiscono, in effetti, il «sentiero» che ci permette di «percorrere», all'interno del nostro sistema, l'itinerario che ci porta

dagli ordini agli articoli ordinati. Dando un'occhiata alla rappresentazione grafica del nostro sistema avremo facilmente un'idea del percorso seguito.

Le due relazioni non sono dello stesso genere: infatti mentre tra ordini e Rigaord sussiste una relazione di tipo 1 a N (per ogni ordine esistono più righe d'ordine), tra Rigaord e articoli sussiste una relazione di tipo 1 a 1 (per ogni riga d'ordine esiste un articolo). Di conseguenza, se l'archivio di lavoro è l'archivio Ordini, l'archivio di ricerca sarà Rigaord (dove il sistema, per ogni registrazione in ordini, cercherà TUTTE quelle collegate), mentre l'archivio Articoli sarà un archivio di decodifica, questa volta rispetto a Rigaord.

Analizzando poi la colonna dBASE, ci accorgiamo che appare una nuova istruzione: SET SKIP TO (in dBASE III non esiste). Se presente, questa attiva la funzione di scorrimento sull'archivio Rigaord: per ogni registrazione di ordini sarà ricercata ogni registrazione di Rigaord collegata (determinando la visua-

```

fig.H  ELABORAZIONE LATO ARCHIVI ANAGRAFICI - SECONDA FASE

SELE 1                                SELECT CCOD,CNOM,ONUM
USE CLIENTI INDEX CL_COD              FROM CLIENTI,ORDINI
USE ORDINI INDEX OR_CODCLI IN 2      WHERE CCOD=OCLI;
USE RIGAORD INDEX RI_NORD  IN 3
SELE 1                                SELECT CCOD,CNOM,ONUM,RORD
SET RELA TO CCOD INTO B              FROM CLIENTI,ORDINI,RIGAORD
SET SKIP TO B,C                      WHERE CCOD=OCLI AND ONUM=RORD;
SELE 2                                SELECT CCOD,CNOM,ONUM,ODAT,RART,RQUA
SET RELA TO ONUM INTO C              FROM CLIENTI,ORDINI,RIGAORD
SELE 1                                FROM CLIENTI,ORDINI,RIGAORD
LIST CCOD,CNOM,B->ONUM                WHERE CCOD=OCLI AND ONUM=RORD;

LIST CNOM,B->ONUM,C->RORD

LIST CCOD,CNOM,B->ONUM,B->ODAT,;
C->RART,C->RQUA

fig.I  ELABORAZIONE LATO ARCHIVI ANAGRAFICI - TERZA FASE

SELE 1                                SELECT CCOD,CNOM,ONUM,ODAT,RART,RQUA,
USE CLIENTI INDEX CL_COD              ADES,APRE,RQUA*APRE
USE ORDINI INDEX OR_CODCLI IN 2      FROM CLIENTI,ORDINI,RIGAORD,ARTICOLI
USE RIGAORD INDEX RI_NORD  IN 3      WHERE CCOD=OCLI AND ONUM=RORD
SELE 1                                AND RART=ACOD;
SET RELA TO CCOD INTO B
SET SKIP TO B,C
SELE 2                                SET RELA TO ONUM INTO C
USE ARTICOLI INDX AR_COD IN 4
SELE 3                                SET RELA TO RART INTO D
SET RELA TO RART INTO D
SELE 1                                LIST CCOD,CNOM,B->ONUM,B->ODAT,;
LIST CCOD,CNOM,B->ONUM,B->ODAT,;    C->RART,C->RQUA,D->ADES,D->APRE,;
C->RQUA*D->APRE

```

Figura 4 - Relazioni multiple lato anagrafici. Le relazioni tra i vari archivi costituiscono quindi una rete di collegamento percorribile da una parte all'altra, per prelevare i vari dati che interessano. Ad esempio la risposta all'interrogazione «quali clienti hanno ordinato l'articolo matita?» utilizza quattro archivi e tre relazioni.

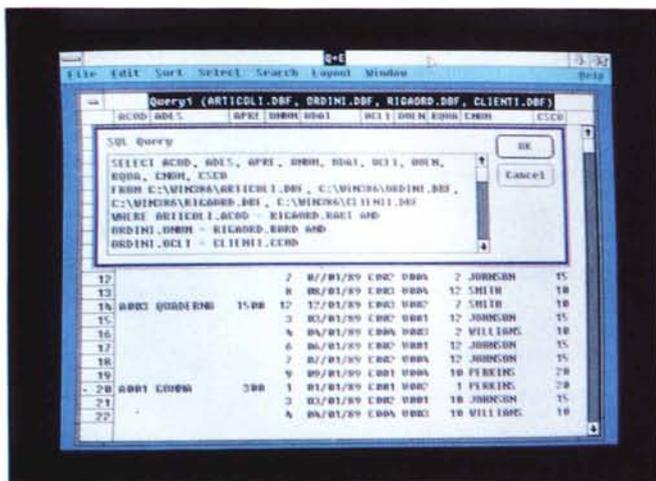


Figura 5 - Excel Q+E. Un paio di numeri fa abbiamo parlato del Q+E, accessorio di Excel, che permette di eseguire delle Query su archivi DBF, e di passarne i risultati in Excel. L'interrogazione può essere anche confezionata in una modalità SQL-like, nel senso che utilizza sintassi SQL e modalità Windows.

lizzazione di una tabella con un numero di righe pari a quello di Rigaord, non più quindi solo la prima riga di ordini; in caso di assenza dell'istruzione SET SKIP TO, la tabella sarà di dimensioni pari a quelle di ordini, poiché il sistema visualizzerà per ogni ordine solo la prima riga d'ordine.

Il funzionamento in SQL è leggermente diverso (vedete subito che non ci sono nuove istruzioni, il procedimento è quello di sempre), ma preferiamo rimandare di qualche riga la sua descrizione.

Figure D,E - Il quarto esempio non presenta nulla di diverso rispetto al secondo, ma gli archivi sono aperti secondo una sequenza differente per far sì che nell'esempio successivo potessero essere aperti anche gli archivi Articoli e Rigaord. Abbiamo così cominciato a ragionare su tutti e cinque gli archivi che compongono il nostro sistema.

Questo può ora essere visto da varie angolazioni, ognuna corrispondente ad un archivio:

— Lato clienti, venditori: lavorando su clienti (o venditori) possiamo, per ogni cliente (o venditore), vedere TUTTI gli ordini in relazione 1 a N, e per ogni ordine il venditore (o il cliente) in relazione 1 a 1; inoltre per ogni ordine potremo richiedere TUTTE le righe d'ordine in relazione ancora una volta 1 a N, e per ogni riga d'ordine l'articolo ordinato, in relazione 1 a 1.

— Lato Ordini: lavorando invece su ordini possiamo, per ogni ordine, vedere TUTTI i clienti e venditori mediante le relazioni 1 a 1; inoltre, come nella situazione precedente, per ogni ordine potremo richiedere TUTTE le righe d'ordine in relazioni questa volta 1 a N, e per ogni riga d'ordine l'articolo ordinato, in relazione 1 a 1.

— Lato Rigaord: lavorando su Rigaord possiamo, per ogni riga d'ordine, risalire agli ordini a cui appartengono (relazioni 1 a 1), dagli ordini risalire ancora ai clienti e ai venditori (relazione 1 a 1). Nell'altra direzione potremo, con una relazione 1 a 1, ottenere le informazioni relative agli articoli.

— Lato Articoli: abbordando il nostro sistema dal lato articoli, possiamo solo risalire la china: da un articolo, in relazione 1 a N, possiamo risalire alle righe d'ordine, da queste, in relazione 1 a 1, agli ordini e dagli ordini, sempre in relazione 1 a 1, ai Clienti e ai Venditori.

Man mano che illustreremo gli esempi risulterà sempre più chiaro che, mentre in dBASE è sempre necessario tener presente posizione degli archivi, relazioni, archivi di lavoro e di decodifica, cioè lo stato nel nostro sistema, estremamente condizionante per il buon esito delle interrogazioni, in SQL tutto ciò è molto meno essenziale, anzi perde quasi del tutto importanza.

Figura F - Questa volta ci spingiamo fino al livello di clienti e venditori, richiamando anche dei valori calcolati su dati provenienti da archivi diversi.

Figure G, H, I - Negli ultimi tre esempi vediamo una elaborazione del lato archivi anagrafici. Lavorando quindi su clienti discendiamo verso ordini sfruttando una relazione 1 a N. Proseguendo verso il basso raggiungiamo Rigaord attraverso ordini sfruttando un'altra relazione 1 a N. Da Rigaord raggiungiamo articoli per mezzo di una relazione 1 a 1.

A questo punto riprendiamo il discorso accennato a proposito del terzo esempio. La sintassi dBASE ci obbliga, in caso di relazione 1 a N, ad utilizzare il comando SET SKIP TO per indicare al sistema una particolare operazione: per ogni riga dell'archivio di lavoro, saranno

visualizzate tutte le righe con essa collegate nell'archivio di ricerca. Se l'istruzione SET SKIP TO viene omessa, il sistema si limita a visualizzare, per ogni riga dell'archivio di lavoro, solo la prima riga ad essa collegata nell'archivio di ricerca.

In SQL la relazione viene definita dall'elenco degli archivi e dall'opzione WHERE seguita da una espressione logica di uguaglianza tra i due campi della relazione.

Utilizzando l'opzione WHERE si ottiene una tabella la cui estensione è pari al numero delle registrazioni presenti in un archivio per il numero di righe dell'altro archivio per le quali la formula logica di relazione è soddisfatta.

Non utilizzando l'opzione WHERE, indicando quindi solo gli archivi che ci interessano, ma non il criterio di relazione, otterremo una tabella N per M, dove il numero di righe è il prodotto del numero di righe dei due archivi.

Questa modalità è già nota a chi conosce il comando Join del dBASE (ma non solo del dBASE).

Conclusioni

Anche il meno esperto leggendo le due colonnine con le due soluzioni, quella dBASE e quella SQL, dello stesso problema, avrà capito che i comandi relazionali sono sempre e comunque pochi.

Ed avrà anche capito che è facile impostare un sistema di relazioni, non tanto quando si conoscono questi pochi comandi, ma soprattutto quando si ha ben chiara in mente l'organizzazione, il disegno, della Banca Dati.

I comandi Relazionali non sono comunque i soli presenti in un linguaggio e quindi la loro padronanza non significa la padronanza del tutto. Sono però quelli fondamentali, ad esempio quando occorra impostare delle Query, ovvero interrogazioni su archivi o su insiemi di archivi.

L'interrogazione, o Query, è il termine che indica genericamente la manipolazione di Banche Dati, finalizzata non a modificarne il contenuto, ma a ricavarne ulteriori informazioni.

Nelle grosse aziende, e i linguaggi tipo SQL sono destinati a queste, ormai il problema non è quello di costituire le Banche Dati, che già esistono in quanto contengono i dati di tutte le procedure operative, quelle che fanno «funzionare» l'azienda, ma quello di sfruttarle il più possibile, delegando questa funzione (lo sfruttamento) ai vari utenti, che in definitiva sono quelli a cui tali ulteriori informazioni servono.

AT 286 16MHz

80286 INTEL 12MHz 0WS 16 operativi
512Kb RAM esp. a 4 Mb Piastra madre
SUNTAC-EMS controller per 2Fd e
2Hd Floppy da 1,2Mb Fujitsu HD da 20
Mb Seagate Tastiera Ita.102 tasti, sche-
da video a scelta , monitor monoc.

TUTTO A LIRE.....1.650.000

386sx 16MHz

80386sx 16MHz 0WS 21 operativi 1Mb RAM Piastra
NEAT-EMS controller per 2FD e 2HD floppy da 1,2 e
HD 20Mb Seagate Scheda video a scelta, Monitor mono-
cromatico, Tastiera 102 tasti italiana 2 seriali 1 parallela

TUTTO LIRE.....2.350.000

AT 286 21MHz

80286 INTEL 16MHz 0WS 21 operativi
1Mb RAM esp. a 4Mb Piastra madre
NEAT-EMS controller 1:1 2FD e 2HD
floppy da 1.2 Fujitsu HD da 20Mb Sea-
gate Tastiera Ita.102 tasti, scheda video
a scelta , monitor monoc. Shadow RAM
per bios (bios AMI) Gestore integrato
per memoria specifiche LIM-EMS , 2
seriali 2 parallele

TUTTO A LIRE.....1.850.000

HD 40Mb veloce + Lire 250.000
FD 1.44Mb Epson + Lire 150.000
su tutti i computers

386 25MHz

80386 INTEL 20MHz 0WS 25 operativi
1Mb RAM , Piastra madre NEAT-EMS
controller 1:1 2Fd e 2Hd , Floppy da 1.2
Fujitsu e HD da 20Mb Seagate , Tastiera
102 tasti Ita. Scheda video a scelta , mo-
nitor monoc. 2 seriali 2 parallele.

TUTTO A LIRE.....2.850.000

386 34MHz

Come sopra ma con 386 INTEL a
25MHz 0 WS 34.1 operativi.

TUTTO A LIRE.....3.300.000

Personal Computer

Asem-Epson-Apple-Compaq
Sharp-Toshiba-Wyse

MiniComputers Honeywell

Stampanti

STAR LC24-10.....L. 599.000
STAR LC24-15.....L. 899.000
NEC P2200.....L. 599.000
NEC P7+.....L. 1.500.000
CITIZEN SWIFT 24.....L. telef.
EPSON (tutti i modelli , telefonare)
HONEYWELL.....L. telef.

STAR LASER 8.....L. 2.899.000
MANNESMAN LASER L. 1.999.000

PLOTTERS

ROLAND DXY-1100.....L. 1.550.000
BENSON-OCE.....L. telef.
HOUSTON INST.....L. telef.

VARIE

VGA 800x600 8bit 256Kb..L. 299.000
VGA 1024x768 16b 512Kb.L. 450.000
Tavola Graf.12x12 Genius..L. 580.000
Mouse Logitek-Microsoft..L. telef
Scanner Genius GS4000.....L. 299.000
Modem GVC.....da L. 150.000

TELEFAX

Murata-Philips-Canon

Canon FAX-80

Formato A4 omologato alimentatore da
5 documenti G3 compatibile 220V 13W
polling, alta risoluzione, massima facilità
d'uso.

OFFERTA Lire 1.300.000

SOFTWARE

DOS-UNIX-XENIX-APPLE

- * Borland
- * Microsoft
- * Samna
- * Digital Research
- * Ashton-tate
- * Lotus
- * Life Boat
- * Ast

Settore DTP

Finesse Logitek.....L. 400.000
PageMaker.....L. telef
Ventura.....L. telef
Corsi di apprendimento

COORDINA

Potente Software di contabilità generale
fatturazione, magazzino gestione ordini
gestione conto corrente
Versione multiutente, tutto in Italiano

386 50MHz

386 INTEL 33MHz 0WS 53.6 operativi
64Kb cache memory 1Mb RAM con-
troller 1:1 2Fd e 2Hd Floppy da 1.2 Fu-
jitsu e HD da 20Mb Seagate , scheda vi-
deo a scelta , monitor monoc. 2 seriali 2
parallele Tastiera 102 tasti Ita.

TUTTO A LIRE.....4.800.000

VGA 800x600 8bit 256Kb + monitor 14"
monocromatico f.b. 16 toni di grigio.
+ Lire.....350.000
A colori..... **+ lire.....750.000**

GESTUDIO

Potente Software per la gestione delle
pratiche dello Studio Legale:
Citazioni , Ricorsi , Esecuzioni, Infortuni
Stragiudiziali , Parcellazione , Statistiche
Scadenario delle udienze
Stesura atti giudiziali

POSSIBILITA DI RATEIZZO A TASSI BASSISSIMI

Condizioni di vendita

Tutti i prezzi si intendono I.V.A. esclusa
Spedizione con corriere in tutta Italia
Garanzia 12 mesi totale

**LOW PRICE
LEADER**

Ware Bit

PRESENTI A
ROMAUFFICIO '90
STAND 23

Viale Pasteur 70, 00144 Roma

Tel. 06/ 591 28 26 - 591 83 07 - 591 94 45 - Fax 592 30 25