

Teoria e Pratica Relazionale

I concetti fondamentali

Stiamo assistendo ad una vera e propria esplosione di prodotti di tipo RDBMS (Relational DataBase Management System), prodotti che servono per costruire applicazioni, prevalentemente di tipo gestionale, che si appoggiano su più tabelle legate tra di loro da regole relazionali.

Per poter utilizzare al meglio tali prodotti è necessario avere una buona conoscenza delle teorie relazionali, altrimenti non si riesce a sfruttare tutte le loro potenzialità (dei prodotti), anzi addirittura si rischia di usarli male fino a rendere inutilizzabile l'applicazione che si realizza.

di Francesco Petroni

Sull'argomento Teoria Relazionale esistono decine di libri, ognuno dei quali in genere si riferisce ad uno specifico «Modello Relazionale», che fa uso di terminologie e di simbolismi grafici spesso in contrasto con quelli proposti per un modello concorrente. Noi evitiamo di buttarci nella mischia proponendo argomenti quanto più possibili semplici e comprensibili e cercando di realizzare subito degli esempi pratici, riferendoci a terminologie comuni (e quindi non tecniche) ed usando, per le necessarie verifiche pratiche, qualche prodotto RDBMS per PC.

Puntualizziamo alcune premesse:

- le regole relazionali esistono indipendentemente dall'esistenza dei prodotti RDBMS, dei computer, dell'uomo,
- fortunatamente la Teoria Relazionale (che quindi come appena detto «esiste in natura») e che come appena detto si esplicita in una serie di Modelli e in una serie Regole Relazionali, è abbastanza semplice ed in molti casi intuitiva, per cui il suo studio non prevede ostacoli concettuali insormontabili,
- tutti i prodotti RDBMS pretendono che l'utilizzatore conosca la teoria relazionale, in quanto, nel momento in cui si esplicitano le relazioni tra le tabelle, fanno uso della terminologia standard, che quindi viene data per nota.

Il nostro obiettivo è quello di affrontare tale teoria, cercando di spiegarla con parole semplici e con esempi pratici, realizzati con vari prodotti.

Il Caso Studio che vi proponiamo è adatto a coprire le principali strutture presenti nella teoria relazionale. Alcune di queste sono molto frequenti, altre, invece, lo sono di meno.

Realizzeremo quindi una serie di tabelle, con pochi campi, in pratica solo quelli che servono per impostare le relazioni, e con pochissimi record, giusto per poter controllare l'esattezza del risultato di certe operazioni.

Non ci occuperemo della costruzione delle strutture, dell'immissione dei dati, né delle problematiche relative al controllo dei campi.

Ci limiteremo ad esaminare solo i controlli incrociati derivanti dal fatto che le tabelle sono in relazione tra di loro.

Il Caso Studio riguarda un Database che inizialmente sarà così organizzato:

- una tabella Numeri, di una rivista che parla di Informatica,
- una tabella Articoli, in relazione con la precedente tramite il campo numero della rivista, in cui l'articolo stesso è stato pubblicato.

Inoltre l'articolo è identificato da un numero progressivo,

– una tabella Autori, in relazione con la precedente.

La relazione è di tipo Uno a Molti con gli autori dal lato uno e gli articoli lato molti,

– una tabella Rubriche, cui fanno riferimento gli articoli. Anche qui la relazione è di tipo Uno a Molti tra rubriche e articoli.

Useremo successivamente una quinta tabella Insiemei quando vorremo, ma lo faremo alla fine, ipotizzare che un articolo possa essere scritto a più mani.

Dapprima costruiremo queste quattro tabelle.

Poi dando per scontato il fatto che le tabelle già sono state costruite e riempite di dati, le maneggeremo sia con il dBase della Borland che con Access 2.0 della Microsoft allo scopo, lo ripetiamo, di analizzare solo gli aspetti relazionali.

Costruiremo poi una piccolissima serie di varianti che ci permetteranno di analizzare dei casi particolari, di più difficile reperimento.

Parleremo infine anche un po' di SQL.

In figura 1 vediamo i dati delle quattro tabelle (più una) e nella successiva le strutture delle stesse in formato dBase.

Un po' di terminologie

Un Database è un insieme di Tabelle collegate tra di loro attraverso dei legami logici chiamati Relazioni. Una relazione si basa sulla corrispondenza tra un campo di una tabella e un campo di un'altra tabella (esistono varianti quando il collegamento avvenga attraverso più campi, o parte di campi).

I tipi principali di relazione sono due:

– relazione Uno a Molti, la più diffusa in assoluto. Es. in un numero di MC ci sono più articoli, uno specifico articolo è pubblicato in un numero di MC, quindi tra numeri e articoli c'è una relazione uno a molti. Es. un autore scrive più articoli, un articolo è scritto da un autore.

– relazione Uno a Uno, meno diffusa. Es. un autore può essere un collaboratore di MC, un collaboratore è tale se è autore.

Esiste teoricamente anche un altro tipo di relazione:

– relazione Molti a Molti.

Questo tipo, che esiste in teoria, per poter essere risolto in pratica si DEVE spezzare in due relazioni Uno a Molti. Ad esempio tra numeri della rivista e autori degli articoli c'è una relazione molti a molti. Infatti un autore può scrivere più articoli nello stesso numero, e su uno stesso numero hanno scritto più autori, quindi tra numeri e autori c'è una relazione Molti a Molti.

La relazione molti a molti si risolve inserendo una terza tabella, in questo esempio è proprio quella degli articoli, che è in relazione uno a molti sia lato numeri (un numero, molti articoli) sia lato autori (un autore, molti articoli).

Quindi se tra le due tabelle c'è una relazione molti a molti è necessario, SEMPRE, inserire una terza tabella intermedia, che in alcune teorie relazionali assume il nome molto chiarificatore di Matrice.

Ad esempio tra le Fatture e gli Articoli, in un'applicazione di tipo gestionale, la relazione è molti a molti (una fattura ha molti articoli, un articolo è presente in molte fatture). La tabella intermedia si potrebbe chiamare dettaglio della fattura (oppure righe) e sta in relazione uno a molti con la fattura (una fattura ha molte righe di dettaglio) e con gli articoli (uno stesso articolo è presente in molte

Figura 1 - Teoria e Pratica Relazionale - I dati su cui lavoreremo.

Eseguiamo una serie di esercizi sviluppati su un Database con quattro Tabelle, ma che poi aumenteremo a cinque. I Numeri di una rivista, gli Articoli apparso nei diversi numeri, gli Autori, scrittori degli articoli e le Rubriche cui i vari articoli appartengono. Le strutture sono molto leggere e contengono solo i campi che servono per impostare le relazioni. Anche i dati sono pochissimi, ma significativi, per eseguire efficacemente le varie prove. Qui vediamo i dati di tutte e quattro le tabelle, più quelli della quinta che vedremo alla fine.

Record	ANUM	APRG	AAUT	ATIT	ARUB
1	100	01	AA	IL PROCESSORE PENTIUM	HW
2	100	02	BB	PROVA LOTUS 123	SW
3	100	03	EE	PROVA AUTOCAD 13	SW
4	100	04	DD	LE API DI WINDOWS	WN
5	100	05	DD	LE MODALITA' GRAFICHE	GR
6	100	06	AA	LE SCHEDE SVGA	HW
7	100	07	CC	I MODEM A 14400 BAUD	TM
8	101	01	AA	COME SCEGLIERE IL CLOCK GIUSTO	HW
9	101	02	CC	I SEGRETI DI INTERNET	TM
10	101	03	BB	DBASE V PER WINDOWS	SW
11	101	04	EE	PROGRAMMARE IN LISP	GR
12	101	05	BB	I NUOVI WORD PROCESSOR	SW
13	101	06	AA	MODEM PER TUTTE LE TASCHE	TM
14	101	07	EE	MODELLAZIONE SOLIDA - TEORIA	GR
15	101	08	AA	I CHIP 64MEGABYTES ?	HW
16	102	01	DD	I NUOVI PRODOTTI MICROGRAF	GR
17	102	02	AA	WINDOWS NT SERVER-WORKSTATION	WN
18	102	03	AA	IL BUS - TIPOLOGIE	HW
19	102	04	DD	PROGRAMMAZIONE VISUALE	SW
20	102	05	BB	ATTERRARE CON FLIGHT SIMULATOR	GM
21	102	06	CC	COLLEGAMENTI REMOTI	TM
Record	RCOD	RDES			
1	GM	GIOCHI			
2	GR	GRAFICA			
3	HW	HARDWARE			
4	SW	SOFTWARE			
5	TM	TELEMATICA			
6	WN	WINDOWS			
Record	SSIG	SNOM	SCOG	SCAP	SQLF
1	AA	ALDO	ALBANI	CC	COLLABORATORE
2	BB	BIAGIO	BELLINI	CC	COORDINATORE
3	CC	CARLO	CECCHETTI	DD	CAPOREDATTORE
4	DD	DARIO	DONINI		DIRETTORE
5	EE	ENZO	ERCOLI	CC	COLLABORATORE
6	FF	FRANCO	FERRARI	BB	COLLABORATORE
7	GG	GIANNI	GALENI	BB	COLLABORATORE
Record	RCOD	RDES			
1	GM	GIOCHI			
2	GR	GRAFICA			
3	HW	HARDWARE			
4	SW	SOFTWARE			
5	TM	TELEMATICA			
6	WN	WINDOWS			
Record	ISIG	INUM	IPRG		
1	DD	101	02		
2	EE	101	02		
3	EE	101	03		
4	FF	101	03		
5	AA	101	05		
6	CC	101	05		
7	BB	102	05		
8	EE	102	05		
9	CC	102	05		

righe di dettaglio, delle varie fatture).

Altre varianti nascono quando occorre individuare le due Tabelle su cui operano le relazioni. Si possono definire due tipi di relazioni più particolari:

– quando esiste una relazione uno a molti all'interno della stessa tabella, siamo di fronte ad una Gerarchia (che vedremo),

– quando esiste una relazione molti a molti tra dati della stessa tabella si ha un'ulteriore tabella detta Matrice chiusa. Questo è l'unico caso che non vedremo.

Se tra due tabelle esiste una relazione le due tabelle assumono dei vincoli reciproci che non possono essere violati. I dati delle due tabelle devono, in altre parole, essere reciprocamente coerenti. Deve essere rispettata la cosiddetta Integrità Referenziale.

Nei primissimi prodotti, es. il dBase

III, era l'utente che doveva preoccuparsi, nel realizzare le maschere di immissione e di modifica dei dati, anche del rispetto dell'integrità referenziale, e se non lo faceva o non ci riusciva, rischiava di introdurre nel suo database dati scorretti.

Ora sono i più moderni RDBMS che assumono su di sé il gravoso compito di eseguire il controllo della correttezza reciproca tra le due tabelle.

Per Integrità Referenziale si intende uno speciale controllo di congruità dei dati ai due lati della relazione. Ad esempio in una relazione uno a molti non si può eliminare un record nella tabella lato uno se esistono dei record collegati nella tabella lato molti (non si può eliminare un autore se esistono suoi articoli), inoltre non si può inserire un nuovo articolo se non di un autore già presente nella tabella autori.

```

Struttura del file : D:\ARTICOLI.dbf
Numero totale record : 21
Data ultima revisione: 18/10/94
Campo Nome campo Tipo Dim Dec
1 ANUM Carattere 3
2 APRG Carattere 2
3 AAUT Carattere 2
4 ATIT Carattere 30
5 ARUB Carattere 2
Totale: 40

Struttura del file : D:\RUBRICHE.dbf
Numero totale record : 6
Data ultima revisione: 18/10/94
Campo Nome campo Tipo Dim Dec
1 RCOD Carattere 2
2 RDES Carattere 10
Totale: 13

Struttura del file : D:\AUTORI.dbf
Numero totale record : 7
Data ultima revisione: 18/10/94
Campo Nome campo Tipo Dim Dec
1 SSIG Carattere 2
2 SNOM Carattere 12
3 SCOG Carattere 12
4 SCAP Carattere 2
5 SQLF Carattere 14
Totale: 43

Struttura del file : D:\RUBRICHE.dbf
Numero totale record : 6
Data ultima revisione: 18/10/94
Campo Nome campo Tipo Dim Dec
1 RCOD Carattere 2
2 RDES Carattere 10
Totale: 13

Struttura del file : D:\INSIEMI.dbf
Numero totale record : 9
Data ultima revisione: 18/10/94
Campo Nome campo Tipo Dim Dec
1 ISIG Carattere 2
2 INUM Carattere 3
3 IPRG Carattere 2
Totale: 8
    
```

Figura 2 - Teoria e Pratica Relazionale - Le strutture devono essere conosciute a memoria. Queste sono le strutture delle nostre cinque tabelle, espresse in formato dBase. I campi chiave e gli indici che permettono di costruire più organizzazioni logiche sui dati vanno, con dBase, costruiti e gestiti a parte. Una delle più grosse differenze tra dBase e i prodotti più moderni è che i campi chiave, identificativi del record, e i campi indice, campi su cui si eseguono operazioni che si vuole veloci, sono definibili direttamente a livello di struttura.

Prima bisognava costruire, con la programmazione, le procedure di controllo reciproco, oggi basta indicare al prodotto RDBMS che si desidera l'integrità referenziale e si può stare sicuri che non verranno accettati dati scorretti.

In figura 3 vediamo appunto il momento della creazione delle relazioni con Access 2.0. Sono ben chiari i collegamenti (nei nostri esempi le relazioni sono sempre tra due campi) e le loro «direzioni». Per ogni linea occorre poi impostare alcune specifiche nell'apposita dialog box (in basso a destra nella figura), quindi: campi su cui si basa il collegamento, il tipo di relazione (i tipi sono due: Uno a Uno e Uno a Molti), poi se si desidera che Access si occupi dell'integrità referenziale (in alcuni casi potrebbe non essere necessario). Altre due possibilità sono Aggiorna in successione e Cancella in successione. Se si sceglie quest'ultima opzione è possibile cancellare un record di una tabella e tutti i suoi collegati sulla tabella correlata (es. se si cancella un autore spariscono tutti i suoi articoli). Aggiorna in successione è ancora più micidiale. Si cambia la sigla di un autore e vengono cambiate tutte le sigle nei record delle tabelle correlate.

Si tratta come evidente di opzioni da «prendere con le molle» nel senso che, se usate male o per sbaglio, possono provocare cancellazioni o modifiche indesiderate. Nella versione 1.1 di Access queste due opzioni non c'erano e quindi non si poteva cancellare un record che aveva dei record correlati in un'altra tabella se non dopo aver cancellato questi ultimi.

Nella box Relazioni esiste anche un pulsante Tipo Join di cui parleremo tra un po'.

Alcune operazioni usando il dBase standard

In un'applicazione dBase si costruiscono più file DBF, ognuno dei quali corrisponde ad una tabella, secondo la terminologia che stiamo utilizzando. Ad ognuno di questi file si possono associare degli indici, che sono anche questi file (desinenza NDX), e che servono sostanzialmente a due cose, a mettere in ordine i dati, secondo l'indice, e a fare sul file, attraverso l'indice, delle ricerche veloci.

Tra due tabelle è possibile, con uno specifico comando, stabilire una relazione. La relazione indica il campo della prima tabella che punta al campo della seconda tabella, che deve obbligatoriamente essere indicizzata su quel campo.

Stabilita così la relazione, quando si scorre la prima tabella viene contestualmente scorsa la seconda tabella, che si posiziona via via sul primo record corrispondente.

In pratica si verificano tre possibilità:
 - che al record della prima tabella corrisponda un solo record della seconda. È il caso in cui la relazione è una Uno a Molti vista dal lato Molti. Nei nostri esercizi è il caso ad esempio della lista degli articoli con l'indicazione della rubrica,
 - che al record della prima tabella corrispondano più record della seconda. È il caso di prima visto dalla parte Uno. In questo caso viene puntato il primo dei record della seconda, per cui occorrerà in qualche modo scorrere gli altri. Nei nostri esercizi è il caso della lista delle rubriche a ciascuna delle quali corrispondono più articoli,
 - che al record della prima non corrisponda alcun record della seconda. In tal caso la seconda tabella si troverà in condizione EOF, fine file, nessun record attivo.

In dBase occorre quindi aprire, in varie aree di lavoro, le varie tabelle con gli indici necessari, occorre stabilire le relazioni, se queste sono sufficienti a risolvere il problema. Se la relazione non basta occorre spostarsi tra le varie aree e cercare il record voluto, poi se i record sono un gruppetto e il file è indicizzato, gli altri record risultano essere in sequenza, per cui è molto facile trovarli.

Un buon utilizzatore di dBase III scrive con una certa facilità le varie istruzioni necessarie a raggiungere lo scopo prefissato.

In figura 4 proponiamo un program-

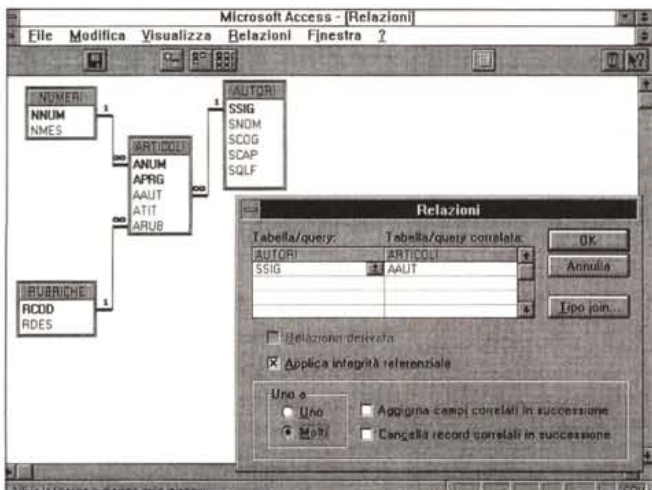


Figura 3 - Teoria e Pratica Relazionale - L'impostazione delle Relazioni con Access 2.0. Per le prove useremo i due prodotti situabili alle estremità della tecnologia RDBMS. Da una parte il dBase III, con il quale costruiremo una serie di programmi che manipoleranno i dati, prelevandoli dalle varie tabelle, dall'altra l'Access 2.0, in cui si lavora su strutture, relazioni, query, ecc. sfruttando al meglio l'interfaccia grafica. Sono in dirittura di arrivo altri prodotti, di pari categoria rispetto ad Access 2.0, dBase V e Paradox 5.0 della Borland, tanto per non fare nomi, e li proveremo anche dal punto di vista relazioni in Access.

lazionale non appena ci arriveranno. Qui vediamo la semplicità con cui si definiscono, graficamente, le relazioni in Access.

ma che esegue quattro elaborazioni differenti, che lavorano sui nostri dati, il cui risultato è mostrato nelle successive quattro figure (dalla 5 alla 8, ma della 8 parliamo dopo).

Mostriamo nella tabella di figura 9 le principali istruzioni utilizzate. Nel listato ci sono molte altre istruzioni che servono in pratica solo per realizzare l'output dei vari sottoprogrammi.

C'è da dire che il dBase III ha avuto molte evoluzioni, sia in termini di servocomandi (l'Assist, il generatore di Query, ecc.) che facilitano la scrittura delle varie istruzioni (ma in fondo ci sono sempre i comandi puntino), sia in termini di nuove istruzioni (es. SET SKIP ON/OFF del dBase IV) che migliorano le operazioni relazionali.

Passiamo ad Access 2.0

In Access 2.0 il lavoro iniziale consiste nel definire le strutture delle varie tabelle (operazione che non vediamo). Rientrano nella definizione della struttura la definizione dei singoli campi e di tutte le loro caratteristiche, ad esempio quale sia il campo chiave (identificativo del record), quali campi si vuole che siano associati ad indice (con o senza controllo dell'univocità), quali siano le eventuali regole di validazione del dato immesso, e così via.

Definite le tabelle occorre definire le relazioni, operazione che si esegue nell'ambiente grafico già visto in figura 3.

A questo punto il database è pronto all'uso, ma prima di usarlo è bene fare una serie di precisazioni:

- in dBase ed in Access sono ovviamente possibili chiavi complesse, ad esempio quelle realizzate combinando due campi. Non le vediamo nei nostri esempi,
- mentre in dBase si creano tabelle ed indici che rimangono fisicamente separati (sono file differenti), in Access tutto il database, tabelle ed indici, dati, e tutto il resto, rientra in un unico fileone (grosso file) con desinenza MDB,
- mentre in dBase è l'utente, o il suo programma, che si deve spostare da una tabella all'altra, da un indice ad un altro, in Access le cose si svolgono automaticamente. Se si ricerca un dato tramite una condizione su un campo e se su quel campo c'è un indice è Access che si posiziona su tale indice. Se si aprono più tabelle si possono usare dati provenienti da tutte le tabelle, in quanto sono le relazioni definite che allineano correttamente i loro record,
- in Access esistono in pratica due tipi di relazioni, quelle fisse, che rimangono attive in ogni situazione, che servono a

```

CLEAR ALL          wait 44 PRIMA ELABORAZIONE
CLEAR
? "NUMERI CON ARTICOLI E RUBRICA"
?
USE NUMERI
SELECT 2
USE ARTICOLI INDEX ARTICOLI
SELECT 3
USE RUBRICHE INDEX RUBRICHE
SELECT 2
SET RELATION TO ARUB INTO C
SELECT 1
DO WHILE .NOT. EOF()
? NNUM
NN=NNUM
SELECT 2
FL=1
SEEK NN
IF .NOT. EOF()
DO WHILE NN=ANUM
IF FL=1
?? " ",APRG,ATIT,ARUB,C->RDES
FL=0
ELSE
? " ",APRG,ATIT,ARUB,C->RDES
ENDIF
SKIP
ENDDO
ENDIF
SELECT 1
SKIP
ENDDO
CLEAR ALL          wait 44 SECONDA ELABORAZIONE
CLEAR
? "LE RUBRICHE E GLI ARTICOLI"
?
USE RUBRICHE INDEX RUBRICHE
SELECT 2
USE ARTICOLI INDEX ARTICOLI2
SELECT 3
USE AUTORI INDEX AUTORI
SELECT 2
SET RELATION TO AAUT INTO C
SELECT 1
DO WHILE .NOT. EOF()
? RCOD,RDES
KK=RCOD
SELE 2
SEEK KK
FL=1
DO WHILE KK=ARUB
IF FL = 1
?? " ",ARUB,ATIT,AAUT,C->SNOM,C->SCOG
FL=0
ELSE
? SPACE(14),ARUB,ATIT,AAUT,C->SNOM,C->SCOG
ENDIF
SKIP
ENDDO
SELECT 1
SKIP
ENDDO
CLEAR ALL          wait 44 TERZA ELABORAZIONE
CLEAR
? "LA GERARCHIA"
?
USE AUTORI INDEX AUTORI
DO WHILE .NOT. EOF()
? SSIG,SCOG,SQLF
SS=SCAP
RP=RECN()
SEEK SS
IF .NOT. EOF()
?? SS,SSIG,SCOG,SQLF
ENDIF
GO RP
SKIP
ENDDO
CLEAR ALL          wait 44 QUARTA ELABORAZIONE
CLEAR
? "ARTICOLI A QUATTRO MANI"
?
USE ARTICOLI INDEX ARTICOLI
SELECT 4
USE RUBRICHE INDEX RUBRICHE
SELECT 3
USE AUTORI INDEX AUTORI
SELECT 2
USE INSIEMI INDEX INSIEMI
SET RELATION TO ISIG INTO C
SELECT 1
SET RELATION TO ARUB INTO D
DO WHILE .NOT. EOF()
KK=ANUM+APRG
AA=ANUM
TT=ATIT
RR=ARUB
SELECT 2
SEEK KK
FL=1
DO WHILE KK=INUM+IPRG
IF FL=1
? AA,TT,RR,D->RDES,ISIG,C->SCOG,C->SNOM
FL=0
ELSE
? SPACE(46),ISIG,C->SCOG,C->SNOM
ENDIF
SKIP
ENDDO
SELECT 1
SKIP
ENDDO
CLEAR ALL

```

Figura 4 - Teoria e Pratica Relazionale - Caro vecchio dBase III.

Tutto ciò che si può ottenere con i nuovi strumenti «servocomandi» presenti nei prodotti delle ultimissime generazioni, lo si poteva già ottenere 10 anni fa usando il dBase III, o dal prompt dell'applicazione, oppure realizzando dei programmini il cui scopo era quello di assemblare una serie di istruzioni, che permettessero sostanzialmente di passare da una tabella all'altra e di scorrere i record di una tabella.

Figura 5 - Teoria e Pratica Relazionale - Elenco dei Numeri e degli Articoli.

Questa e le prossime tre figure costituiscono il risultato delle quattro elaborazioni previste in sequenza dal programma di cui abbiamo visto il listato. La prima mostra i vari numeri con il dettaglio degli articoli del numero e delle rubriche cui gli articoli appartengono. Uno a Molti tra Numeri e Articoli, e tra Rubriche e Articoli. Ce la caviamo con due cicli DO WHILE... END DO, uno dentro l'altro. Quello esterno scorre i numeri, quello interno scorre gli articoli, a parità di numero. La rubrica è fornita dal comando SET RELATION, che permette in pratica di vedere come fossero campi dell'articolo anche i campi delle rubriche (perché ci si trova sul lato Molti della relazione).

NUMERI CON ARTICOLI E RUBRICA

100	01 IL PROCESSORE PENTIUM	HW HARDWARE
	02 PROVA LOTUS 123	SW SOFTWARE
	03 PROVA AUTOCAD 13	SW SOFTWARE
	04 LE API DI WINDOWS	HW WINDOWS
	05 LE MODALITA' GRAFICHE	GR GRAFICA
	06 LE SCHEDE SUGA	HW HARDWARE
	07 I MODEM A 14400 BAUD	TH TELEMATICA
101	01 COME SCEGLIERE IL CLOCK GIUSTO	HW HARDWARE
	02 I SEGRETI DI INTERNET	TH TELEMATICA
	03 DBASE V PER WINDOWS	SW SOFTWARE
	04 PROGRAMMARE IN LISP	GR GRAFICA
	05 I NUOVI WORD PROCESSOR	SW SOFTWARE
	06 MODEM PER TUTTE LE TASCHE	TH TELEMATICA
	07 MODELLAZIONE SOLIDA - TEDIUA	GR GRAFICA
	08 I CHIP 64MEGABYTES ?	HW HARDWARE
102	01 I NUOVI PRODOTTI MICROGRAF	GR GRAFICA
	02 WINDOWS NT SERVER-WORKSTATION	HW WINDOWS
	03 IL BUS - TIPOLOGIE	HW HARDWARE
	04 PROGRAMMAZIONE VISUALE	SW SOFTWARE
	05 ATTERREARE CON FLIGHT SIMULATOR	GH GIOCHI
	06 COLLEGAMENTI REMOTI	TH TELEMATICA

LE RUBRICHE E GLI ARTICOLI			
GN GIOCHI	GM ATTERRARE CON FLIGHT SIMULATOR	BB BIAGIO	BELLINI
GR GRAFICA	GR LE MODALITA' GRAFICHE	DD DARIO	DOMINI
	GR PROGRAMMARE IN LISP	EE ENZO	ERCOLI
	GR MODELLAZIONE SOLIDA - TEORIA	EE ENZO	ERCOLI
	GR I NUOVI PRODOTTI MICROGRAF	DD DARIO	DOMINI
HW HARDWARE	HW IL PROCESSORE PENTIUM	AA ALDO	ALBANI
	HW LE SCHEDE SUGA	AA ALDO	ALBANI
	HW COME SCEGLIERE IL CLOCK GIUSTO	AA ALDO	ALBANI
	HW I CHIP 64MEGABYTES ?	AA ALDO	ALBANI
	HW IL BUS - TIPOLOGIE	AA ALDO	ALBANI
SW SOFTWARE	SW PROVA LOTUS 123	BB BIAGIO	BELLINI
	SW PROVA AUTOCAD 13	EE ENZO	ERCOLI
	SW I NUOVI WORD PROCESSOR	BB BIAGIO	BELLINI
	SW DBASE V PER WINDOWS	BB BIAGIO	BELLINI
	SW PROGRAMMAZIONE VISUALE	DD DARIO	DOMINI
TM TELEMATICA	TM I MODEM A 14400 BAUD	CC CARLO	CECCHETTI
	TM I SEGRETI DI INTERNET	CC CARLO	CECCHETTI
	TM MODEM PER TUTTE LE TASCHE	AA ALDO	ALBANI
	TM COLLEGAMENTI REMOTI	CC CARLO	CECCHETTI
UN WINDOWS	UN LE API DI WINDOWS	DD DARIO	DOMINI
	UN WINDOWS NT SERVER-WORKSTATION	AA ALDO	ALBANI

Figura 7 - Teoria e Pratica Relazionale - La gerarchia negli Autori. Supponiamo che un Autore abbia una sua caratterizzazione (collaboratore, coordinatore, caporedattore e direttore) e abbia (eccetto il direttore) un superiore. Quindi nella struttura della tabella abbiamo sia il codice dell'autore (che è il campo indice principale, altrimenti detto Chiave), sia il codice del suo superiore. Questo codice è in relazione con il campo chiave della stessa tabella. In dBase III, per vedere l'elenco degli Autori con il dettaglio dei loro capi, siamo costretti a scorrere una volta la tabella, e, poi individuo per individuo, a cercare nella stessa tabella il superiore. Per tornare al punto di partenza si può memorizzare il numero del record al quale si era arrivati (lo fornisce la funzione RECNO()).

LA GERARCHIA			
AA ALBANI	COLLABORATORE	CC CC CECCHETTI	CAPOREDATTORE
BB BELLINI	COORDINATORE	CC CC CECCHETTI	CAPOREDATTORE
CC CECCHETTI	CAPOREDATTORE	DD DD DOMINI	DIRETTORE
DD DOMINI	DIRETTORE		
EE ERCOLI	COLLABORATORE	CC CC CECCHETTI	CAPOREDATTORE
FF FERRARI	COLLABORATORE	BB BB BELLINI	COORDINATORE
GG GALENI	COLLABORATORE	BB BB BELLINI	COORDINATORE

ARTICOLI A QUATTRO MANI			
101 I SEGRETI DI INTERNET	TM TELEMATICA	DD DOMINI	DARIO
		EE ERCOLI	ENZO
101 DBASE V PER WINDOWS	SW SOFTWARE	EE ERCOLI	ENZO
		FF FERRARI	FRANCO
101 I NUOVI WORD PROCESSOR	SW SOFTWARE	AA ALBANI	ALDO
		CC CECCHETTI	CARLO
102 ATTERRARE CON FLIGHT SIMULATOR	GM GIOCHI	BB BELLINI	BIAGIO
		EE ERCOLI	ENZO
		CC CECCHETTI	CARLO

passiamo in una situazione Molti a Molti, che va obbligatoriamente gestita con una tabella intermedia. Noi l'abbiamo chiamata Insieme.

far funzionare il database, e che sono quelle che si definiscono con il comando Modifica Relazioni (visto in figura 3). Se ne possono creare altre «dinamiche», di tipo usa e getta, all'interno di una Query semplicemente tracciando una linea tra i campi su cui creare la corrispondenza. Usa e getta perché vivono solo all'interno della Query.

Un altro aspetto che a questo punto è bene puntualizzare riguarda i tipi di controllo eseguiti da Access:

- controllo sul tipo di campo. Se si è definito un tipo di campo, es. data, oppure

numero, può essere inserito solo un dato coerente con il tipo,

- controllo sulla maschera di immissione. È possibile assegnare una «picture» come caratteristica del campo che quindi funziona come filtro in fase di inserimento. Ad esempio il codice fiscale che ha 6 caratteri alfabetici, due numerici, ecc., può essere sottoposto a questo tipo di filtro,
- controllo eseguito dalle regole di validazione. Ad un campo è possibile associare una regola di validazione. Ad esempio una regola per un campo nu-

Figura 6 - Teoria e Pratica Relazionale - Elenco delle Rubriche e degli Articoli. In questo secondo caso si scorre la tabella lato Uno della relazione. Per ogni elemento lato Uno (in pratica per ogni Rubrica) si cerca il primo articolo di quella rubrica e si scorrono solo gli articoli con lo stesso codice di rubrica (che sarebbero i Molti). Anche in questo caso arricchiamo l'articolo con informazioni prese da un'altra tabella. Quelle degli Autori.

merico potrebbe essere «Between 100 And 200»,

- controllo sulla chiave. Se un campo è definito come chiave oppure è come campo indice senza duplicazioni risulterà impossibile inserire un record che abbia la stessa chiave di un record già inserito,
- controllo referenziale. È il controllo, citato prima, che verifica la congruenza dai due lati di una relazione,
- controllo da macro o programma. Nella scheda che serve per l'alimentazione delle tabelle è infine possibile inserire delle procedure (o Macro o procedure classiche, che si costruiscono nell'ambiente Modulo) che vengono eseguite al verificarsi di un evento, tipicamente l'evento «su disattivazione» ovvero all'uscita dal campo.

Prepariamo una Query

Eseguito il lavoro di preparazione di tabelle e relazioni, il successivo lavoro è assolutamente facilitato. Nella figura 10 vediamo come si costruisce una Query, in pratica, attivato il comando Query Nuovo, viene proposto l'elenco delle tabelle (e delle altre query) utilizzabili. Se si scelgono tabelle già legate da relazioni, tali relazioni appaiono anche graficamente. A questo punto si possono scegliere i campi desiderati prendendoli da qualsiasi tabella, in quanto è Access che sfrutta i «sentieri» tracciati dalle relazioni per prelevare i dati corrispondenti.

Anche qui facciamo una serie di precisazioni:

- esistono più tipi di query. Quella che vediamo è quella più usata, la Query di Selezione. Le altre non le vediamo, anche se sono altrettanto interessanti,
- è possibile, ma anche questo non lo vediamo, scegliere campi, imporre criteri di selezione, inserire campi calcolati, definire regole di raggruppamento e di ordinamento, ecc.,
- in caso di necessità, come detto, è possibile definire delle relazioni «volanti»,
- Access salva la Query secondo la sintassi SQL (lo vediamo in figura 11). Il comando SQL può essere copiato ed incollato da altre parti, ad esempio in una macro o in una procedura, per associarlo ad un pulsante, oppure in una applicazione Visual Basic 3.0 che anch'esso riconosce SQL e lavora su file MDB.

I tipi di Join

È possibile definire il tipo di Relazione in due modi differenti.

Il primo modo parte dalla finestra

Figura 9 - Teoria e Pratica Relazionale - Tabella con i principali comandi dBase da usare in problematiche relazionali.

ISTRUZIONI dBASE	SIGNIFICATO
CLEAR ALL	chiude tutte le tabelle, ed altro, es. le variabili.
USE XXXX INDEX YYYY	apre un file e un suo indice.
SELECT Y	sposta l'area di lavoro attiva.
SEEK X	cerca nella tabella attiva, su cui è attivo un indice, il primo record la cui chiave sia X.
SET RELATION TO X INTO Y	stabilisce una relazione tra il campo X della tabella attiva e la tabella aperta nell'area di lavoro Y. Questa deve essere stata preventivamente aperta e deve essere indicizzata sul campo di corrispondenza.
Y->nomecampo	sintassi per utilizzare un campo del record attivo di una altra area di lavoro.
DO WHILE .NOT. EOF() ... SKIP ENDDO	struttura di programmazione che serve per far scorrere tutta la tabella fino al verificarsi della condizione di file file (si verifica oltre l'ultimo record).
DO WHILE X=Y ... SKIP ENDDO	scorre dal record attivo (trovato con SEEK) fino quando X=Y, in genere fino a quando la chiave della tabella attiva è uguale a quella che si sta cercando. Questo metodo si usa quando si scorre il versante molti di una relazione Uno a Molti.

Relazione, in cui c'è un pulsante specifico (tipo di Join), il secondo modo si attiva eseguendo un doppio clic sulla linea che evidenzia la relazione tra due tabelle.

Appare una box, il cui contenuto è personalizzato nel senso che cita direttamente le due tabelle oggetto della relazione, che mostra tre varianti:

- 1 - per far apparire solo i dati che rispettano la corrispondenza (sono presenti in tutte e due le parti),
- 2 - per far apparire anche i dati della prima che non hanno corrispondenti nella seconda,
- 3 - per far apparire anche i dati della seconda che non hanno corrispondenti nella prima.

Se eseguiamo tali varianti con le nostre tabelle Autori ed Articoli:

- nel primo caso vediamo tutti gli articoli, con i loro autori,
- nel secondo caso vediamo tutti gli articoli, con i loro autori, più tutti gli autori che non hanno articoli,
- il terzo caso corrisponde al primo, ma solo perché l'imposizione della integrità referenziale impedisce di avere articoli senza autori.

Se si volesse l'elenco degli autori che non hanno scritto articoli (oppure in una tabella Clienti analizzare i clienti che non hanno fatto ordini) basta inserire un criterio «Is Null» in un qualsiasi campo della tabella correlata. Lo si può fare nella variante 2. In Access «Is Null» significa campo vuoto, oppure non trovato se si sta «pescando» in una tabella correlata.

La gerarchia

Supponendo che gli autori appartenano ad una categoria (per ipotesi e dal

basso verso l'alto: collaboratore, coordinatore, capo redattore, direttore) e che ogni autore di una categoria sia sottoposto ad un altro autore di categoria superiore (non necessariamente), può nascere il problema di gestire e/o di visualizzare questa gerarchia.

Da un punto di vista relazionale la tabella autori è in relazione Uno a Molti con se stessa: un capo ha più sottoposti, un sottoposto ha un solo capo. Ma capi e sottoposti risiedono nella stessa Tabella. Il problema si può facilmente risolvere creando una query che utilizza più volte la stessa tabella e tracciando a mano le linee che rappresentano i collegamenti.

Nel nostro esempio, ben chiarito dalla figura 13, abbiamo aperto quattro volte la tabella Autori, e abbiamo collegato il campo SCAP, codice del capo, della

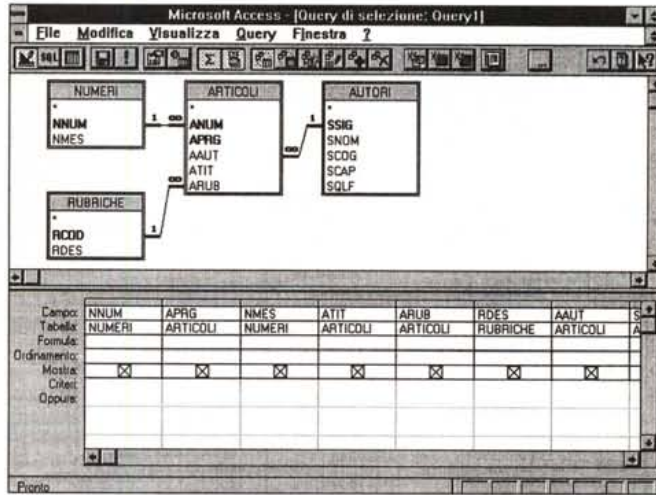


Figura 10 - Teoria e Pratica Relazionale - Come si prepara una Query in Access 2.0. In Access 2.0 le tappe del lavoro sono ben delineate. Prima si costruiscono le Tabelle poi si definiscono le Relazioni.

A questo punto si possono immettere i dati, che vengono controllati sia dalle regole di validazione interne alla tabella, definite a livello di struttura, sia dalle regole di validazione dovute all'aver opzionato il rispetto della Integrità Referenziale.

Non ci si deve quindi più occupare del controllo reciproco dei dati tra le due tabelle.

Quando si costruisce una Query le relazioni definite appaiono fisicamente rappresentate da linee.

Figura 11 - Teoria e Pratica Relazionale - Il lato SQL della Query.

Access 2.0 memorizza la query, impostata dall'utente nell'ambiente QbE, nel formato standard SQL. Il comando così creato, in questo caso si tratta di una query che lavora e visualizza i dati di tutte e tre le tabelle, può essere visto in una sua finestra, in cui può essere scritto, da chi preferisse farlo, invece di confezionarlo con QbE. Il comando si può anche copiare per farlo eseguire ad esempio in una procedura attivata da un tasto, o per essere trasferito in Visual Basic che è anche questo in grado di eseguire comandi SQL.

```

SELECT DISTINCTROW NUMERI.NNUM, NUMERI.NMES, ARTICOLI.ATIT,
ARTICOLI.ARUB, RUBRICHE.RDES, ARTICOLI.AAUT, AUTORI.SCOG
FROM RUBRICHE INNER JOIN (AUTORI INNER JOIN (NUMERI INNER JOIN ARTICOLI
ON NUMERI.NNUM = ARTICOLI.ANUM) ON AUTORI.SSIG = ARTICOLI.AAUT) ON
RUBRICHE.RCOD = ARTICOLI.ARUB;
    
```

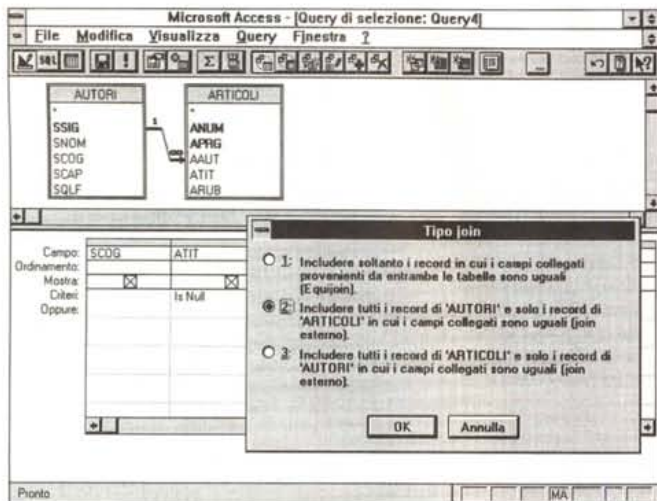


Figura 12 - Teoria e Pratica Relazionale - Alla ricerca del... lavativi. Il tipo di relazione più diffusa è la Uno a Molti. Ad esempio quella tra i nostri Autori e i loro Articoli, in cui un autore ha più articoli mentre un articolo è scritto (per ora) da un solo autore. Se è stato impostato il rispetto dell'integrità referenziale non possono esistere articoli scritti da autori non memorizzati nella tabella. Possono esistere invece autori cui non corrispondono articoli (nei nostri esempi sono Ferrari e Galeni). Si possono facilmente trovare sfruttando una variante della relazione.

seconda con il campo chiave della prima, SCAP della terza con il campo chiave della prima, e così via. Stiamo in pratica utilizzando la prima come tabella dei Direttori (per fortuna ce ne è uno solo), la seconda come tabella dei redattori, e così via.

La freccia sulla linea significa che abbiamo scelto un tipo di Join che mostra comunque i dati della seconda tabella di ogni relazione (variante 3).

Il risultato mostrato nella finestra è chiarissimo e non necessita di ulteriori chiarimenti.

La Uno a Uno, ma che cosa è

Esistono due soli tipi di relazione. Il primo, l'Uno a Molti, è il più diffuso e lo abbiamo visto nelle sue numerose sfaccettature. Il secondo, che è molto meno diffuso, è l' Uno a Uno. Lo spieghiamo con un esempio. Supponiamo che l'insieme dei nostri autori sia costituito da due sottoinsiemi, i giornalisti interni alla rivista, detti Redattori e i Collaboratori esterni, e supponiamo che di questi due sottoinsiemi sia necessario memorizzare informazioni differenti. Occorrerà in pratica costruire necessariamente altre due tabelle, quella dei Redattori e quella dei Collaboratori, ciascuna delle quali è in rapporto Uno a Uno con quella degli Autori. Per il rispetto dell'integrità referenziale un collaboratore deve già essere memorizzato come autore mentre un autore può essere un collaboratore ma può anche non esserlo.

In altre parole l'insieme degli Autori si divide in due sottoinsiemi, Collaboratori e Redattori.

Access 2.0 permette di definire questo secondo tipo di relazione con le stesse modalità con le quali si definisce il primo, e anche da un punto di vista grafico la visualizza correttamente (fig.14).

Il vincolo è che esista un campo di corrispondenza e che questo campo sia chiave di ambedue le tabelle.

Per quanto riguarda l'utilizzo della Uno a Uno, è del tutto analogo alla Uno a Molti, come evidenziato dalla query mostrata in figura 15, in cui elaboriamo dati provenienti dalle tre tabelle (per l'occasione abbiamo alleggerito la struttura di quella degli autori, in quanto alcuni dei dati ora risiedono sulle altre due tabelle).

In sostanza vogliamo realizzare un elenco di Autori con dettagli provenienti dalla tabella Redattori, se l'autore è un redattore, e con dettagli provenienti dalla tabella Collaboratori, se l'autore è un collaboratore esterno.

Anche in questo caso, non essendo-

Figura 13 - Teoria e Pratica Relazionale - La gerarchia completa. Se si suppone che tra gli autori esista una gerarchia (ad esempio, dal basso verso l'alto: collaboratore, coordinatore, capo redattore, direttore) può nascere il problema di gestire e/o di visualizzare questa gerarchia. Da un punto di vista relazionale la tabella autori è in relazione Uno a Molti con se stessa; un capo ha più sottoposti, un sottoposto ha un solo capo. Ma capi e sottoposti risiedono nella stessa Tabella.

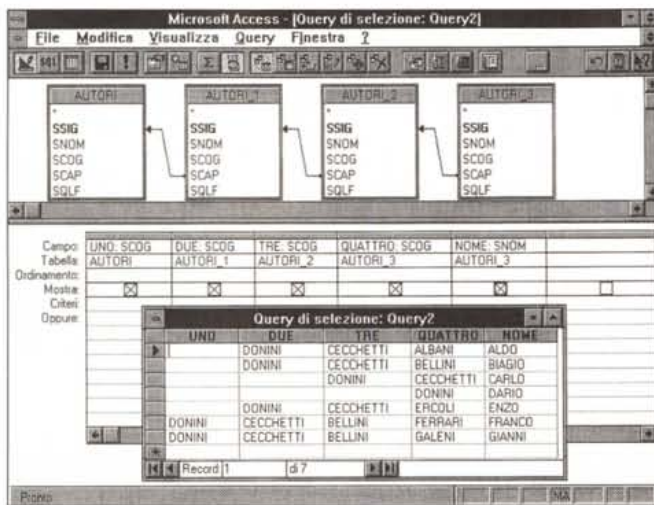
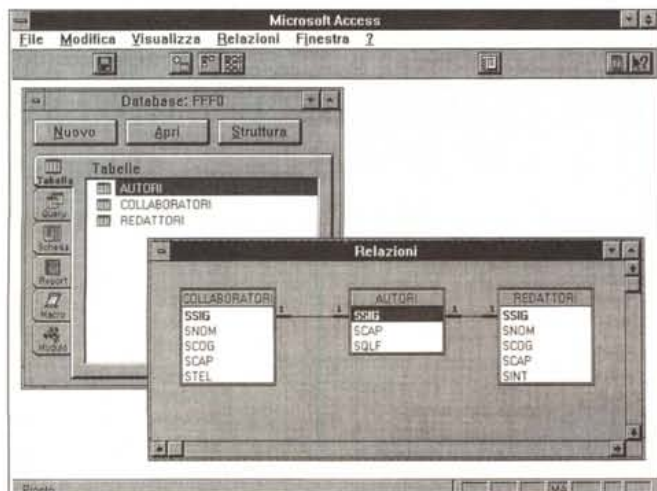


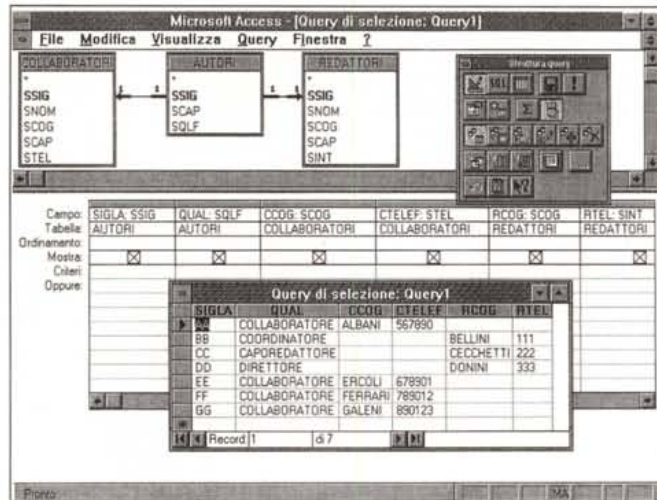
Figura 14 - Teoria e Pratica Relazionale - La relazione Uno a Uno - Definizione. La seconda tipologia di relazione è la Uno a Uno, la spieghiamo con un esempio. Supponiamo che l'insieme dei nostri autori sia costituito da due sottoinsiemi, i giornalisti interni alla rivista e i collaboratori esterni. Se di questi sottoinsiemi è necessario memorizzare informazioni differenti occorre costruire altre due tabelle, quella degli interni e quella degli esterni, ciascuna delle quali è in rapporto Uno a Uno con quella degli autori. Per il rispetto dell'integrità referenziale un collaboratore



tore DEVE già esistere come autore mentre un autore può essere anche un collaboratore.

Figura 15 - Teoria e Pratica Relazionale - La relazione Uno a Uno - Utilizzo.

Nella figura precedente abbiamo visto come appaiono in Access due relazioni uno a uno. Ora le utilizziamo per costruire un elenco di autori con i dati dei collaboratori, se l'autore è un collaboratore, o con i dati degli interni, se si tratta di giornalisti interni. Poiché un interno non può essere esterno i tipi di relazione debbono consentire la visualizzazione dei dati che non corrispondono in tutte e tre le tabelle. Si deve usare una variante del Join.



ci (e non potendoci essere) corrispondenza tra i dati di tutte e tre le tabelle, ma solo corrispondenza a due a due, occorre scegliere un tipo di relazione che consenta la visualizzazione degli autori anche se non sono collaboratori e degli autori anche se non sono redattori.

Infine risolviamo una Molti a Molti

Uno degli errori più frequenti, che si compiono quando si costruisce un'applicazione, consiste nello sbagliare una relazione (ad esempio il suo tipo) e questo, se l'applicazione viene già utilizzata, è un errore gravissimo e spesso irrimediabile.

Supponiamo di avere già costruito la nostra applicazione in cui esistono una tabella Autori e una tabella Articoli. Tra

autori ed articoli c'è una Uno a Molti, ma solo se un articolo è scritto sempre da un solo autore. Se questo non è vero in quanto ogni tanto, magari rarissimamente, un articolo viene scritto da due autori, la relazione diventa Molti a Molti: un autore più articoli, un articolo più autori.

Allora: quando si analizza il database su cui si basa un'applicazione si deve fare, su ogni relazione, una specie di esame di coscienza per capire a quale tipo di relazione ci si trova di fronte. Se è di tipo Molti a Molti può sempre essere molto ben esplicitata con una matrice (fig. 16).

Nel caso in esame, ovvero un articolo scritto da più autori, il rapporto tra autore e articolo fa nascere una terza tabella, che chiamiamo Insieme, in cui c'è l'identificativo dell'autore e l'identificativo dell'articolo (più eventuali altre

MATRICE TRA NUMERI E AUTORI

Autori	Numeri			
	100	101	102	103
Albani	*			*
Bellini	*	*	*	
Cecchetti	*	*		*
Donini		*		*
Ercoli	*			*
Ferrari		*	*	
Galenì			*	

Figura 16 - Teoria Relazionale - Relazione Molti a Molti.

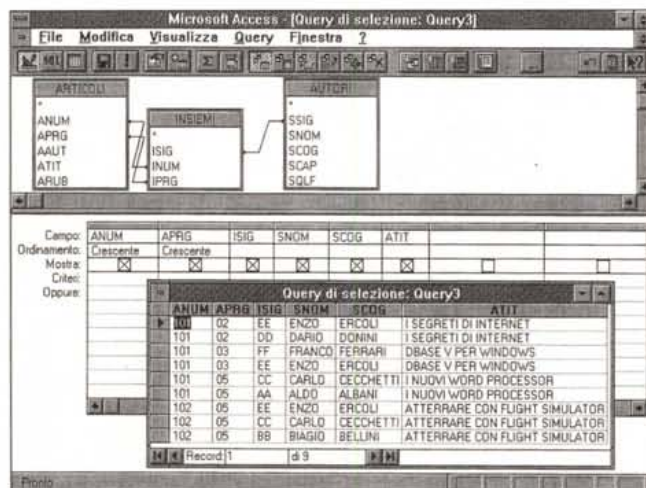
La rappresentazione grafica di una relazione Molti a Molti è una matrice, che tralaltro chiarisce molto bene le cose. Ad esempio, ipotizzando che un autore possa scrivere un solo articolo per numero, possiamo affermare che un autore scrive su molti numeri, mentre su un numero scrivono molti autori. Quindi Molti a Molti. Dal punto di vista pratico una relazione Molti a Molti si risolve SEMPRE creando una tabella intermedia, che la spezza in due relazioni Uno a Molti. Nel nostro esempio è chiaro che la tabella intermedia è quella degli Articoli. Un articolo è scritto da un autore, è pubblicato in un numero, e poi ha sue caratteristiche specifiche, come il titolo o la rubrica cui appartiene. Questa schematizzazione diventa sbagliata nell'ipotesi che un autore scriva più articoli nello stesso numero.

informazioni non significative dal punto di vista relazionale). Ad uno stesso autore possono corrispondere più articoli, ad un articolo più autori.

Nella Query Access che esplora tale ulteriore caso (fig. 17) viene anche evidenziata la relazione tra articoli ed insieme, basata su due campi. Nella finestra vediamo a galla l'elenco degli articoli con i vari autori. È chiaro che questo non impedisce agli autori di scrivere articoli da soli. I questo caso c'è un solo record nella tabella Insieme. In altre parole una relazione Molti a Molti comprende, per così dire, anche i casi Uno a Molti.

Figura 17 - Teoria e Pratica Relazionale - Gli articoli scritti a quattro mani.

In prima approssimazione, lo abbiamo precisato anche nel testo, abbiamo anche supposto che un articolo abbia un solo autore, il che corrisponde ad una relazione Uno a Molti tra autori e articoli. Ma se ora ipotizziamo che l'articolo possa essere scritto da più autori la relazione diventa Molti a Molti (un articolo più autori, un autore più articoli). Va creata «per forza» una tabella intermedia. La nostra la chiamiamo Insieme.



Conclusioni

L'obiettivo dell'articolo era quello di spingere gli utilizzatori del prodotto RDBMS allo studio delle problematiche relazionali, che vanno padroneggiate concettualmente, e di verificare che tutti i prodotti, dal dBase III in poi, in varie maniere, riescono a fronteggiarle e quindi a risolverle praticamente.

Per quanto riguarda i modelli relazionali e le relative problematiche, li abbiamo analizzati quasi tutti, e abbiamo constatato come risolverle con dBase III e con Access 2.0. Ci riproveremo di provare a risolvere gli stessi problemi anche con gli altri prodotti. Sarà anche l'occasione per un ripasso degli aspetti teorici.