

# Teoria Relazionale

## Capire le Relazioni Molti a Molti

*L'enorme diffusione dei prodotti di tipo DBMS anche tra i cosiddetti utenti finali, che in quanto tali non hanno nessuna conoscenza della Teoria Relazionale, sta comportando (è la mia personale opinione) qualche problema.*

*I prodotti DBMS delle ultime generazioni sono ormai molto evoluti, al punto che permettono di costruire applicazioni di media o grande complessità senza che sia necessario scrivere nessun tipo di programma. È facile costruire le Tabelle, le Interrogazioni, le Maschere per l'immissione e la gestione, controllata e guidata, dei dati, i Report per la loro riproduzione, nelle più svariate forme, su carta. È anche facile costruire gli automatismi (quelli di tipo più semplice si chiamano Macro) che semplificano l'esecuzione dei processi ripetitivi*

**di Francesco Petroni**

L'utente dal canto suo conosce (si spera) benissimo il problema che vuole risolvere e quindi conosce bene i dati che deve gestire con l'applicazione che sta costruendo. Magari sa maneggiare bene il prodotto anche perché questo assomiglia molto, in parecchie situazioni operative, a tutti gli altri prodotti che già conosce. Poi abbiamo già detto che non è richiesta nessuna forma di programmazione, specialmente nel caso in cui l'utente sia contemporaneamente quello che realizza l'applicazione e quello che la utilizzerà una volta terminata.

Il cerchio, tra l'utente e il prodotto DBMS, sembrerebbe chiudersi perfettamente, solo che c'è ancora un piccolo problema. Purtroppo, al nostro utente finale, manca una cosa fondamentale, senza la quale l'applicazione che sta realizzando probabilmente non potrà mai funzionare. Manca la conoscenza della Teoria Relazionale, teoria che costituisce il pilastro portante dei prodotti DBMS e che deve essere ben padroneggiata e ben applicata nella fondamentale fase iniziale della costruzione del Database su cui si poggia l'applicazione.

In altre parole, prerequisito per il buon funzionamento dell'applicazione, è la corretta impostazione del Database, ovvero la definizione corretta delle varie Tabelle coinvolte e delle Relazioni che le legano l'una alle altre.

In genere i manuali dei vari prodotti di cui stiamo parlando dedicano comunque qualche pagina alla descrizione dei vari modelli relazionali, ma tali descrizioni sono «addomesticate», nel senso che la trattazione è finalizzata all'ap-

prendimento del prodotto e non della Teoria Relazionale, che ha una sua valenza assoluta, che prescinde dai prodotti, dal computer, dall'uomo. La teoria relazionale esiste in natura, così come la legge della Gravitazione Universale oppure il Secondo Principio della Termodinamica. L'uomo si è solo accorto della loro esistenza.

### Uno a Molti, Uno a Uno e Molti a Molti e... basta

La Teoria Relazionale, qualsiasi sia il modello formale che si utilizza (alcuni studiosi hanno messo a punto dei Mo-

delli Teorici che sfruttano anche dei semplici simbolismi grafici, un po' come si fa in altri tipi di progettazione, ad esempio nella progettazione degli Impianti Elettrici) si basa su poche tipologie di elementi. In particolare tra due delle molte Tabelle presenti in genere in un Database possono esistere solo tre tipi di Relazione: la Relazione Uno a Molti, la Relazione Uno a Uno, la Relazione Molti a Molti.

Nell'analizzare due tabelle collegabili tra di loro ci si deve quindi subito porre una domanda: in che tipo di rapporto sono? Successivamente, al momento della realizzazione dell'applicazione, a

*Figura 1 - Il concetto di Matrice.*

*Una relazione Molti a Molti si presta ad essere simboleggiata in forma di Matrice rettangolare. Nel nostro caso mostriamo una Matrice tra Articoli, in alto, e Fatture, a sinistra. La matrice, letta in verticale, indica in quale fattura è stato venduto ciascun articolo. Letta in orizzontale indica quali articoli sono stati venduti in una data fattura. La casella in corrispondenza di una fattura e di un articolo può indicare la quantità di quell'articolo venduta nella fattura stessa. Questa informazione è propria della Tabella Righe della Fattura. Il prezzo di vendita è una informazione che è propria della tabella Articoli. Se si ipotizza di definire degli sconti in maniera non predeterminata (ad esempio se lo Sconto è proprio del Cliente) occorrerà prevedere un prezzo di vendita specifico per quella singola riga oppure, sempre nella riga, uno sconto relativo proprio a quell'articolo.*

Articolo	TORRONE	SPUMANTE	GIANDUIA	PANETTONE	PANFORTE	CIOCCOLATA	PANEPATO	CHAMPAGNE	PANDORO
Fattura									
00001		6		15		5		5	
00002						5	5		
00003			120						
00004				10	30				
00005	10							50	
00006	20					20			
00007			20		15		11		6
00008		12						12	
00009		5	6			10			
00010	15			5			5		

seconda della risposta che si dà alla domanda e a seconda del prodotto che si ha a disposizione, occorre fare una serie di operazioni, in genere sempre le stesse, indipendenti dalla particolare applicazione che si sta affrontando.

In altre parole, nello sviluppo di un' applicazione di tipo DBMS, è molto più critica la fase preliminare di Analisi del Problema, nella quale vanno definite le Tabelle e le Relazioni tra di esse, che non la fase attuativa nella quale si creano le strutture delle Tabelle, delle Interrogazioni, in cui si costruiscono le Maschere e i Report. Se non funziona la struttura relazionale non funziona l'intera applicazione, mentre se non funziona una Maschera o un Report, basta correggerli.

Torniamo all'argomento dell'Articolo (relazioni Molti a Molti) e quindi alla trattazione degli aspetti relazionali facendo una serie di esempi per sensibilizzarvi alla materia. Prima però anticipiamo il fatto che, in caso di relazione Molti a Molti, tra due Tabelle occorrerà sempre individuare, e conseguentemente costruire, una terza Tabella, intermedia tra le due, che spezza la relazione Molti a Molti in due relazioni Uno a Molti.

Altra premessa doverosa riguarda il fatto che la corretta impostazione delle Tabelle e delle Relazioni è sì fondamentale per il corretto funzionamento dell'applicazione, ma non è tutto. Occorrerà comunque eseguire altre impostazioni, inserire altri controlli, oltre a quelli impliciti nel fatto di avere definito le Regole Relazionali. In particolare è fondamentale il discorso sul Campo Chiave e sui Campi Indice da inserire nelle Strutture delle varie Tabelle, e che servono, tra le altre cose, a far funzionare le Relazioni. Di questo e di alcuni altri controlli parleremo via via che capiterà.

### Ecco un po' di esempi

Descriviamo un'applicazione di tipo Biblioteca. Un Autore (escludiamo il caso di libri scritti da più autori) ha scritto uno o più Libri (inteso come Titolo), un dato Libro è stato scritto da un solo Autore. Un Libro, scritto dal suo Autore, può avere più Edizioni, magari di Case Editrici differenti.

Tra Autori e Libri c'è una relazione Uno a Molti. Anche tra Libri ed Edizioni c'è una Uno a Molti. Se invece pensiamo anche alle Case Editrici diremo che tra Case Editrici e Titoli c'è una Molti a Molti. Dato un Titolo, ad esempio un classico, può essere stato edito da più Case Editrici e data una Casa Editrice ha edito più Titoli. Per la soluzione di

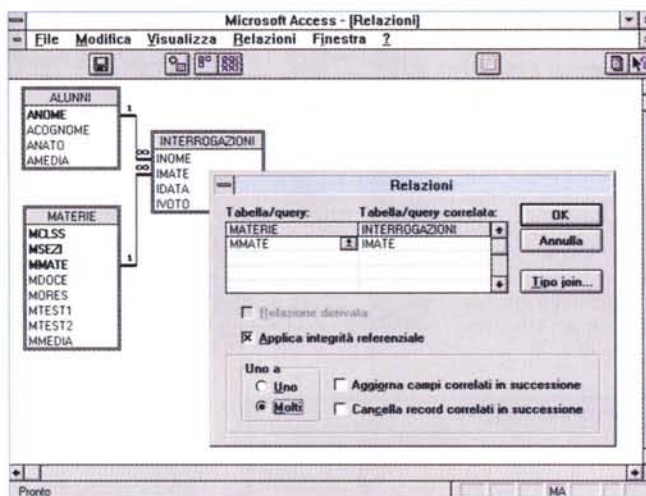


Figura 2 - Il nostro primo Caso Studio: Alunni, Materie Scolastiche e Interrogazioni. Schema Relazionale. Ecco un'esemplificazione facilmente comprensibile del concetto di Relazione Molti a Molti, tipo di relazione che «esiste in natura», ma che con qualsiasi prodotto, di tipo DBMS, può essere risolta solo costruendo tre Tabelle. Poiché tra una Materia Scolastica e un Alunno esiste un rapporto Molti a Molti, va costruita una Tabella intermedia che, in questo caso, si può chiamare Interrogazioni. Un Alunno «subisce» molte interrogazioni nelle varie materie. In una

Figura 3 - Il nostro primo Caso Studio: Alunni, Materie Scolastiche ed Interrogazioni. Prima Scheda.

Questa è una Scheda con Sottoscheda, realizzata con MS Access, che mostra in alto i dati «anagrafici» della Materia e in basso i dati relativi alle interrogazioni dei singoli Alunni. La scheda può essere utilizzata anche per eseguire l'immissione dei dati. Sulle righe della sottoscheda vengono eseguiti dei calcoli poi riportati in alto. Si tratta del calcolo del numero di interrogazioni registrate e della media dei voti riportati. Usiamo i voti da 0 a 10, interi. Si potrebbe prevedere una scheda, simmetrica a questa, che mostra i dati delle Interrogazioni non dal punto di vista della Materia, ma da quello dell'Alunno.

NOME	COGNOME	NATO	DATA	VOTO
FRANCO	ARANCIONE	10/12/80	27/09/95	6
BRUNO	VERDI	09/09/80	29/09/95	7
MARIO	ROSA	25/08/80	01/10/95	6
LUIGI	NERI	30/04/80	04/10/95	5
MICHELE	ROSSI	21/10/80	05/10/95	8
MARIO	ROSA	25/08/80	07/10/95	8
MASSIMO	COBALTO	05/06/79	09/10/95	6
PAOLO	AZZURRI	11/09/79	09/10/95	10
MARCO	VIOLA	01/01/80	09/10/95	4
MASSIMO	COBALTO	05/06/79	11/10/95	7
MARCO	VIOLA	01/01/80	11/10/95	9

Figura 3 - Il nostro primo Caso Studio: Alunni, Materie Scolastiche ed Interrogazioni. Prima Scheda. Questa è una Scheda con Sottoscheda, realizzata con MS Access, che mostra in alto i dati «anagrafici» della Materia e in basso i dati relativi alle interrogazioni dei singoli Alunni. La scheda può essere utilizzata anche per eseguire l'immissione dei dati. Sulle righe della sottoscheda vengono eseguiti dei calcoli poi riportati in alto. Si tratta del calcolo del numero di interrogazioni registrate e della media dei voti riportati. Usiamo i voti da 0 a 10, interi. Si potrebbe prevedere una scheda, simmetrica a questa, che mostra i dati delle Interrogazioni non dal punto di vista della Materia, ma da quello dell'Alunno.

questo tipo di rapporto, che esiste in natura e che viene chiamato Matrice (vedi figura 1), occorre prevedere una Tabella Intermedia che si chiama proprio Edizioni e che avevamo già identificato.

In questo caso è abbastanza facile intuire la necessità della tabella intermedia, in altri casi no. La Tabella Edizioni conterrà le due informazioni necessarie per il collegamento alle due tabelle Titoli e Editore, più altre proprie, come anno di edizione, prezzo di copertina, ecc.

Parliamo ora di Alunni e Materie Scolastiche, in mezzo ci sono le... Interrogazioni. Un Docente di una Materia

Scolastica fa tante Interrogazioni, anche più di una per Alunno. Un Alunno viene interrogato più volte nella stessa Materia.

Per quanto riguarda il Docente, se ci interessa solo conoscere il nome, lo inseriremo come campo Docente nella Tabella Materia. Se invece ci interessa gestirlo (ad esempio perché con la nostra applicazione trattiamo anche le Classi e gli Orari) dovremo «dedicargli» una Tabella tutta sua.

Parliamo ora della nostra Rivista, per la quale scrivono parecchi Autori. Un Numero di MC ha tanti Articoli (Uno a Molti), un dato Autore ha scritto tanti

Articoli (altra Uno a Molti). Anche in questo caso tra Numeri di MC e Autori c'è una Matrice risolvibile con la Tabella intermedia Articoli. L'articolo ha un suo titolo, un suo numero di pagine, ecc. oltre che nome dell'autore e numero di MC in cui è apparso, campi necessari per costruire i collegamenti.

Parliamo anche dei Lettori della nostra Rivista. Si possono dividere in due gruppi: gli Abbonati e i non Abbonati. Tra l'insieme dei Lettori di MC e l'insieme degli Abbonati c'è una Relazione Uno a Uno. Il che, parlando di Lettori e di Abbonati, significa che: dato un Let-

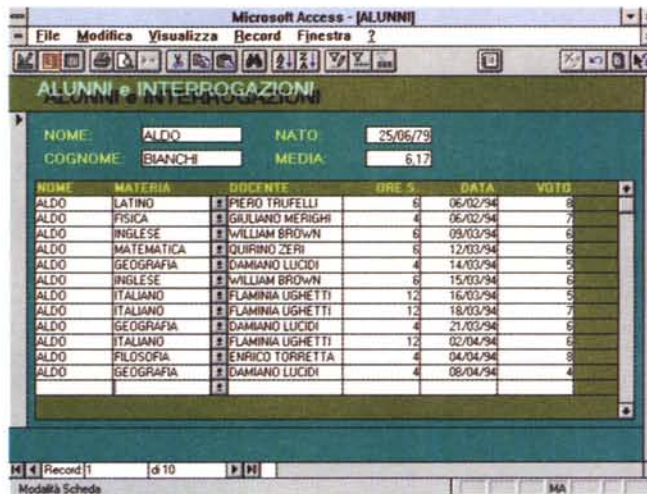


Figura 4 - Il nostro primo Caso Studio: Alunni, Materie Scolastiche ed Interrogazioni. Seconda Scheda. Una relazione Molti a Molti, che prevede una tabella in mezzo, può avere due punti di osservazione. In certi casi uno dei due è più importante, ad esempio se si parla di Fatture ed Articoli è più interessante vedere una Fattura con le sue Righe di Dettaglio, che non un Articolo con le Righe delle varie Fatture in cui è stato venduto. Nell'applicazione Alunni e Materie è opportuno vedere le interrogazioni da ambedue le parti.

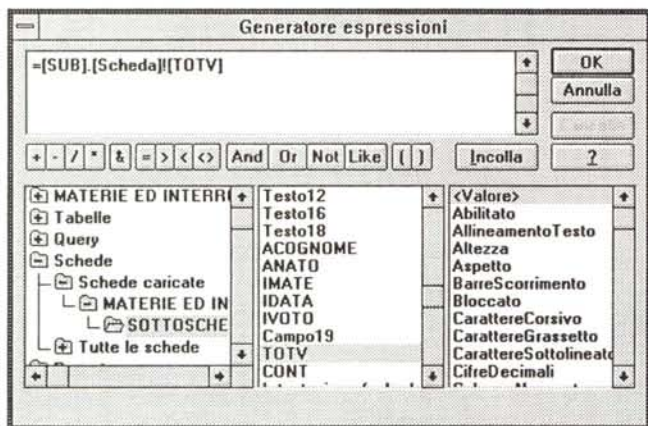


Figura 5 - Il nostro primo Caso Studio. Generazione di una formula. Nella nostra scheda abbiamo inserito una Casella di Testo che mostra la media delle votazioni di tutte le interrogazioni. Il calcolo è stato eseguito nella sezione Più di Pagina della Sottoscheda, in una Casella di Testo che è stata chiamata TOTV e della quale è stata impostata a No la proprietà Visibile. Nella Casella di Testo della soprascheda va inserita la proprietà che ne indica l'origine, origine che è la Casella di Testo della sottoscheda, che si chiama SUB (è il nome con cui la sottoscheda è vista dalla scheda).

relazione Molti a Molti. Una Fattura contiene tanti Articoli, un Articolo è presente in tante Fatture. La tabella intermedia sarà quella delle Righe di Dettaglio, che conterrà sue informazioni, ad esempio la quantità ordinata.

### Perché usiamo MS Access

Nei costruire i nostri esempi utilizzeremo MS Access versione 2.0, in quanto è il prodotto DBMS oggi più in voga e in quanto è «corretto» dal punto di vista relazionale, nel senso che permette di sviluppare tutte le tipologie di Modelli Relazionali e che permette di impostare, come controlli eseguiti dal sistema, tutte le Regole di Integrità Referenziale.

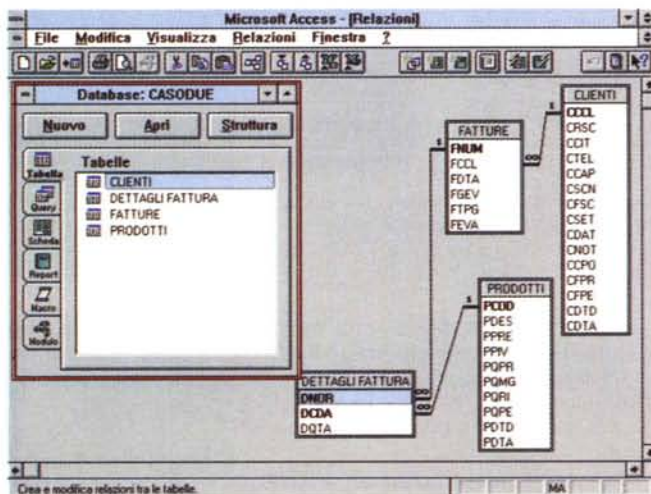
Vedremo quindi, per ognuno dei tre Casi Studio che proporranno, la vista grafica delle Relazioni (comando di Menu: Modifica Relazioni), ambiente in cui vanno letteralmente disegnate, come normali linee tra un campo di una tabella e l'altro dell'altra tabella, le Relazioni.

Altro strumento caratteristico di MS Access, e che lo rende molto adatto a mostrare i dati presenti in Tabelle poste in relazione Uno a Molti e conseguentemente in relazione Molti a Molti, è la Scheda con Sottoscheda.

In pratica nella parte superiore della Scheda si vedono i dati dalla tabella «lato Uno» e nella parte inferiore i dati della tabella «lato Molti». Nella Molti a Molti, che come detto si spezza in due Uno a Molti (e quindi diventa Uno a Molti a Uno): in alto vedremo la tabella lato Uno (una delle due) e in basso la tabella intermedia Molti arricchita con i dati dell'altra tabella lato Uno (la rimanente).

La Molti a Molti è simmetrica, nel senso che si possono creare due Schede con Sottoscheda simmetriche tra di loro, nel senso che mostrano in pratica la stessa sottoscheda, vista da due sopraschede differenti. Vedremo, nel primo Caso Studio, Interrogazioni Scolastiche organizzate sia per Materia che per

Figura 6 - Il nostro secondo Caso Studio: Fatture e Articoli - Schema Relazionale. In una Fattura vengono venduti più Articoli. Un Articolo viene venduto in più Fatture. Tra le due tabelle va costruita una terza tabella che contiene le Righe di Dettaglio delle Fatture, ciascuna delle quali indica che in quella Fattura viene venduto quell'Articolo. Appartengono alla tabella Righe di Dettaglio altri campi, ad esempio la Quantità di quell'articolo venduto in quella fattura. La fattura sarà a sua volta collegata alla Tabella dei Clienti. In questo caso la relazione è di tipo Uno a Molti. Un Cliente molte Fatture, una Fattura un sol Cliente. Nella figura abbiamo piazzato sia la finestra Database che mostra le quattro Tabelle, sia, in secondo piano, la finestra Relazioni.



tore di MC può essere un Abbonato di MC. Un Abbonato di MC è comunque un Lettore di MC.

Parliamo infine di una classica appli-

cazione Ordini e/o Fatture. Ci sono i Clienti e ci sono la Fatture (Uno a Molti). Poi ci sono gli Articoli ordinabili. Tra Fatture ed Articoli c'è chiaramente una

Figura 7 - Il nostro secondo Caso Studio: Fatture e Articoli - Una scheda.

La tipologia di scheda più adatta per «vedere» una Relazione Molti a Molti è la Scheda con Sottoscheda, ormai facilmente realizzabile con qualsiasi prodotto DBMS della ultima generazione. Facilmente realizzabile a patto che si padroneggi la sottostante problematica relazionale. Nella parte superiore della Scheda si inseriscono i dati di una delle due Tabelle Molti, nella parte inferiore, i dati della tabella intermedia arricchita con i dati della altra Tabella Molti. Nel primo caso, visto prima, in alto le Materie e in basso le Interrogazioni con dati degli Alunni, oppure in alto gli Alunni ed in basso le Interrogazioni con dati delle Materie. Nel caso di Fatture ed Articoli: in alto i dati della Fattura, in basso le Righe di Dettaglio con in più dati provenienti dalla Tabella Articoli. È possibile anche inserire una serie di campi calcolati, che si poggiano sulla stessa organizzazione relazionale.

Microsoft Access - [IMMISSIONE FATTURE]

File Modifica Visualizza Record Finestra ?

**IMMISSIONE FATTURE**

NUMERO: 00018      SETTORE: 4

CLIENTE: BIANCHI MARIO      AREA GEOGR.: MILANO

TIPO PAGAMENTO: CONTANTI      INIZIO RAPPORTO: 05/04/90

DATA: 15/04/93      Settore di Attività

GIORNI: 30       Artigianato     Commercio     Industria     Servizi     Altro

EVASO IL: 09/05/93

SCONTO: -2%      604.900    104.831    709.731

ARTICOLO	PREZZO	% IVA	QTA'	NETTO	TOTIVA	LORDO
SPUMANTE	8.900	19%	2	17.800	3.362	21.162
GIANDUIA	3.600	9%	10	36.000	3.240	39.240
PANETTONE	11.000	9%	15	165.000	14.850	179.850
PANFORTE	2.700	19%	8	21.600	4.104	25.704
CIOCOLATA	1.300	9%	10	13.000	1.170	14.170
PANPEPATO	4.500	19%	5	22.500	4.275	26.775
CHAMPAGNE	37.000	19%	10	370.000	70.300	440.300

Record 18 di 25

Modaltà Scheda

Alunno.

Altro strumento di Access che sfrutteremo molto è la Query che permette di creare delle Tabelle «virtuali» con i dati, anche provenienti da più tabelle, organizzati nella maniera più adatta alle necessità dell'applicazione. Le query producono dei Dynaset, che sono viste dirette sui dati delle tabelle, e che sono quindi direttamente aggiornabili dalla query stessa. Nel creare una Scheda con Sottoscheda si possono usare, come Origine dei Dati, sia tabelle che query.

## Il primo Caso Studio

Il primo Caso Studio che vogliamo affrontare è quello delle Interrogazioni, che prevede la costruzione di tre tabelle: quella degli Alunni, quella delle Materie che, per semplificare, arricchiamo di informazioni relative a Docenti e a Libri di Testo. Non essendo un'applicazione critica non cediamo alla tentazione di dare un Codice Identificativo né alle Materie, né agli Alunni. Gli Alunni li identificheremo attraverso il Nome (sperando che non ve ne siano due con lo stesso nome, nel tal caso dovremmo inserire anche l'iniziale del Cognome) e anche le Materie attraverso il loro nome. In pratica definiremo questi due campi come Campi Chiave nelle rispettive tabelle.

È infatti necessario disporre, sia nella tabella Alunni che in quella Materie, di un campo chiave che serve proprio per impostare le due relazioni Uno a Molti con la terza tabella Interrogazioni. Questa conterrà innanzitutto i due campi, nome dell'Alunno e nome della Materia, che costituiscono il collegamento con le rispettive tabelle (potrebbe essere un buon aiuto guardare la figura 2, in cui vediamo le due tabelle, in secondo pia-

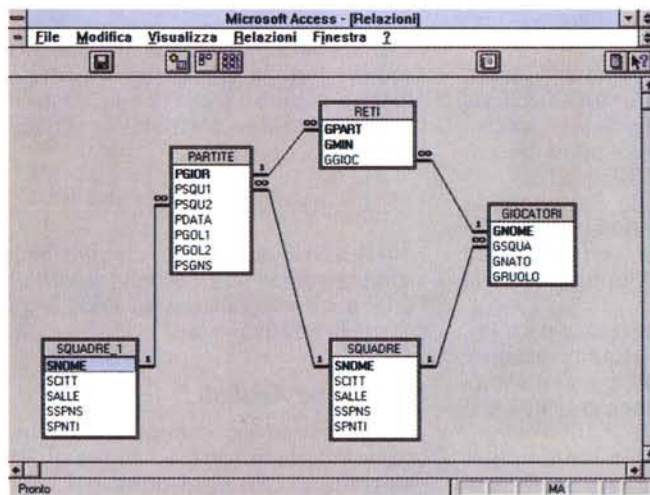


Figura 8 - Il nostro terzo Caso Studio: Squadre di Calcio, Partite di un Campionato, Calciatori e Gol. Schema Relazionale. In un Campionato di Calcio una Squadra incontra tante altre Squadre. La tabella intermedia, tra le due tabelle delle Squadre (che invece sono la stessa tabella e da un punto di vista relazionale possono esserlo) si può chiamare Partite. Questa contiene altre informazioni prime tra tutte il Risultato dell'incontro. Se gestiamo anche i Giocatori, che sono in rapporto Uno a Molti con le Squadre, potremmo memorizzare anche i Gol segnati nelle varie partite. Tra i Giocatori e le Partite c'è la tabella intermedia dei Gol: un Giocatore tanti Gol e una Partita tanti Gol. Si tratta di una Molti a Molti da manuale. Nella tabella intermedia dei Gol metteremo Partita, Giocatore, Minuto, e magari tipo di gol, se ci va di memorizzare questo tipo di informazione.

no, e, in primo piano, una fase della definizione di una delle due relazioni).

Nella Tabella Interrogazioni non è indispensabile, per il corretto funzionamento dell'applicazione, impostare una Chiave identificativa. Inoltre l'accoppiata Alunno+Materia non potrebbe essere utilizzata come Chiave identificativa perché non è univoca: un Alunno può essere interrogato più volte nella stessa Materia. Allora si potrebbe pensare ad Alunno+Materia+Data dell'interrogazione. Questa scelta potrebbe andare bene per mettere in ordine i dati e funziona a patto che un povero Alunno non venga interrogato due volte nella stessa Materia nello stesso Giorno.

Nella figura 2 vediamo due cose. Sullo sfondo le tre tabelle con le due relazioni Uno a Molti. In primo piano la Box

nella quale si definiscono i dettagli delle relazioni. I due campi in comune, tra la tabella Materie e la tabella Interrogazioni, sono il nome della Materia (MMATE e IMATE, nelle due tabelle).

È in questo momento che si imposta lo switch «Applica integrità referenziale», il che significa che Access impedisce che siano inserite Interrogazioni in materie non presenti nella relativa Tabella e che siano cancellate Materie per le quali siano state registrate Interrogazioni (a condizione che non sia attivato lo Switch «Cancella in successione»).

Definite Tabelle e Relazioni si può passare alla costruzione delle Schede. Ne facciamo due (in figura 3 e 4), di tipo Scheda con Sottoscheda, in cui vediamo Materie e Interrogazioni di quella

particolare Materia e Alunni ed Interrogazioni di quel particolare Alunno.

La prima sottoscheda mostra i dati delle Interrogazioni e degli Alunni (preparate da una Query), la seconda i dati delle Interrogazioni e delle Materie (preparate da una Query).

Si tratta di una costruzione abbastanza semplice, peraltro guidata da un Wizard, ottimo per preparare una prima bozza della Scheda, sulla quale poi intervenire a mano.

## Elenco dei passaggi necessari per costruire Scheda con Sottoscheda

Supponendo di avere già costruito le tre Tabelle necessarie in un Modello Relazionale Molti a Molti (tabelle che

chiameremo UNO1 - MOLTI - UNO2) e supponendo di avere già impostato le relazioni tra UNO1 e MOLTI e UNO2 e MOLTI, elenchiamo i passi necessari per costruire, con MS Access, la scheda con sottoscheda di figura 3 o, i passi sono gli stessi, di figura 4 o di figura 7.

1 - Costruzione di una Query con le due tabelle MOLTI e UNO2, su cui basare la sottoscheda;

2 - costruzione, con l'Autocomposizione, di una Scheda Superiore di tipo colonna singola, basata sulla tabella UNO1;

2.1 - eventuali interventi interni alla sovrascheda, indipendenti dall'aspetto relazionale;

3 - costruzione, con l'Autocomposizione, della Sottoscheda di tipo tabellare, basata sulla QUERY;

3.1 - eventuali interventi interni alla sottoscheda, indipendenti dall'aspetto relazionale;

3.2 - inserimento di eventuali Caselle di Testo in cui eseguire dei calcoli sui dati della sottoscheda (totalizzazioni, conteggi, ecc.) ed impostazione del loro nome (ad es. TOT1, TOT2, ecc.);

4 - inserimento della Sottoscheda nella Scheda superiore ed impostazione del nome (ad. es. SUB), con cui l'intero oggetto sottoscheda viene visto nella scheda superiore;

5 - collegamento tra le due parti impostando come valori «Campi master» e «Campi figli» rispettivamente i campi di collegamento tra la tabella UNO1 e la QUERY;

5.1 - inserimento di eventuali elementi che visualizzano, nella sovrascheda, calcoli eseguiti dalla sottoscheda.

Per quanto riguarda i passi 3.1 e 3.2 si tratta di interventi di tipo «estetico» o comunque di tipo «localizzato». Ad esempio nei nostri esempi abbiamo sostituito delle semplici Caselle di Testo con delle Combo Box. Invece di scrivere il nome di un Alunno lo scegliamo da una Lista di nomi.

Inoltre tra i passi indicati con 3.2 prevediamo la costruzione di elementi che eseguono dei calcoli su colonne della sottoscheda. Ad esempio la Media dei voti di tutte le interrogazioni presenti nella sottoscheda. Tale elemento, in genere si tratta di una Casella di Testo, si può rendere invisibile, se ci interessa solo il suo risultato, magari da visualizzare in un'altra casella di testo. Così nella sovrascheda inseriamo una Casella di Testo la cui origine è una formula (costruibile a botta sicura con il Generatore di Espressioni, in figura 5) che fa semplicemente riferimento alla casella nella sottoscheda. Notare il percorso che permette di raggiungere la casella di testo... agognata (Schede, Schede carica-

```
Function RUOLO (NUM)
' scodifica del codice numerico
' indicante il Ruolo di ciascun giocatore
Select Case NUM
Case 1
  RUOLO = "PORTIERE"
Case 2
  RUOLO = "DIFENSORE"
Case 3
  RUOLO = "CENTROCAMPISTA"
Case 4
  RUOLO = "MEZZAPUNTA"
Case 5
  RUOLO = "PUNTA"
End Select
End Function

Function SEGNO (G1, G2)
' calcolo del segno della schedina
' a partire dal numero di gol di ciascuna squadra
SEGNO = IIf(G1 > G2, "1", IIf(G1 = G2, "X", "2"))
End Function
```

rico, da 1 a 5, che indica, nella Tabella.Giocatori, il ruolo svolto da quel particolare giocatore.

te, Materie ed Interrogazioni, Sottoscheda Interrogazioni, nome del campo, proprietà Valore). Dato che si lavora in famiglia, in pratica nella stessa scheda, la sintassi si semplifica:

=[SUB]![SCHEDE]![TOTV]

In cui SUB è il nome assunto dalla Sottoscheda all'interno della scheda e TOTV è il nome della casella con il calcolo nella Sottoscheda.

### Fatture ed Articoli

Una classica applicazione in cui si trova una relazione Molti a Molti è quella che tratta di Fatture e di Articoli. Una Fattura tanti Articoli, un Articolo in tante Fatture. In mezzo ci va una tabella intermedia, quella con le Righe di Dettaglio della Fattura. Dalla parte delle Fatture abbiamo piazzato anche la tabella Clienti. Uno a Molti: un Cliente molte Fatture, una Fattura un solo Cliente.

In figura 6 vediamo lo schema Relazionale, insieme alla Finestra Database che mostra le quattro tabelle.

La scheda, in figura 7, è al solito una Scheda con Sottoscheda ed è molto ben sviluppata. Nella parte superiore dati della Fattura e dei Clienti, nella inferiore Righe e Articoli, più una serie di campi calcolati. I dati della sottoscheda sono stati preparati, come al solito, da una query.

La scheda serve per inserire o per modificare contestualmente i dati della Fattura e i dati delle Righe. I dati dei Clienti e degli Articoli appaiono solo per controllo.

Sia al momento della scelta del Cliente cui fare la fattura, che al momento della scelta dell'Articolo cui si riferisce la Riga, si possono utilizzare delle Com-

Figura 9 - Il nostro terzo Caso Studio. Un paio di utili Function.

In questo articolo parliamo degli aspetti Relazionali tra le tabelle dell'applicazione. Tabelle e Relazioni costituiscono l'ossatura fondamentale dell'applicazione, che però necessita di tante altre cosette per poter funzionare al meglio. In ogni applicazione che si rispetti vanno costruite delle Function da utilizzare qui e là quando necessario, ad esempio in una Query, in una Scheda, in un Report. Nell'applicazione Calcio potranno tornarci come una funzione che «calcola» il segno della Schedina dati i gol segnati dalle due squadre in una partita. Oppure una funzione che scodifica un codice nume-

rico, da 1 a 5, che indica, nella Tabella.Giocatori, il ruolo svolto da quel particolare giocatore.

bo Box che mostrano, rispettivamente, la lista dei clienti e la lista degli articoli. Poiché non è possibile, con tale scheda, inserire né un nuovo Cliente né un nuovo Articolo va prevista una pulsantiera, con bottoni del tipo di quelli messi in alto nella scheda in figura, che apra la Scheda Anagrafica dei Clienti (o degli Articoli) in una situazione Record Nuovo. Altri servizi da... bottonizzare saranno la Stampa della Fattura, l'uscita dalla Scheda, ed altri.

In questa applicazione sono stati molto sviluppati i campi calcolati. Le caselle con i vari calcoli sono stati piazzati nel Piè di Pagina della sottoscheda e sono richiamati dalle rispettive caselle nella sovrascheda.

### Il campionato di Calcio

Quasi tutti gli italiani conoscono il Campionato di Calcio, con le sue squadre, i suoi giocatori, le sue partite, i suoi gol. Dei milioni di italiani che parlano del campionato sono pochissimi, fortunatamente, quelli che lo hanno osservato dal punto di vista relazionale. Lo facciamo anche noi, limitando la nostra trattazione al campionato di un solo anno e parlando solo di Squadre, Giocatori, Partite e Gol (potete cominciare a vedere la figura 8). Non parliamo di Formazioni, non parliamo di Arbitri e meno che mai di Segnalinee, non parliamo di vari Campionati ma solo di uno.

Tra Squadra e Squadra ci sono gli Incontri. Una squadra incontra tante altre squadre, e anche la seconda ne incontra tante altre. Ci troviamo quindi di fronte ad una classica Relazione Molti a Molti, con la piccola variante che le due tabelle in relazione sono in pratica la stessa tabella (la struttura relazionale si chiama Matrice Chiusa). La tabella inter-

Figura 10 - Il nostro terzo Caso Studio: Squadre di Calcio, Partite di un Campionato, Calciatori e Gol. Scheda a Tre Livelli.

In questa Scheda vediamo tre livelli di dati. In alto le Squadre, poi, nella sottoscheda posta in mezzo, per ogni squadra i suoi incontri, sia quelli giocati in casa che quelli giocati fuori casa. Posizionandosi su uno degli incontri, in basso viene attivata un'ulteriore sottoscheda che mostra i gol di quell'incontro, segnati da una qualsiasi delle due squadre. L'aspetto tecnico più interessante è quello che riguarda la produzione della sottoscheda in cui vengono mostrate le partite giocate dalla squadra, sia in casa che fuori. La sottoscheda si poggia su una Query (è la sua Origine dei Dati).

The screenshot shows a Microsoft Access window titled "SQUADRE - PARTITE - GOL". It features three sub-forms:

- Top Form (Squadre):** Fields include NOME (INTER), ALLENATORE (HODGSON), PUNTI (21), CITTA' (MILANO), and SPONSOR (PIRELLI).
- Middle Form (Partite):** A table with columns: GIORNATA, SQUADRA1, SQUADRA2, DATA, GOL1, GOL2, SEGNO. Data rows show matches between INTER, MILAN, JUVENTUS, LAZIO, and TORINO.
- Bottom Form (Gol):** A table with columns: PARTITA, MINUTO, GIOCATORE, SQUADRA, RUOLO. It lists goal scorers like ROBERTO CARLOS, SIGNORI, GANZ, WINTER, and INCE.

The screenshot shows a Microsoft Access window titled "Query di selezione: QUERY PARTITE DI UNA SQUADRA". It displays a list of fields in the "PARTITE" table and a grid for defining the query's criteria and fields.

Campo:	PGIOR	PSQU1	PSQU2	PDATA	PGOL1	PGOL2	SEGN0	SEGN1	PGOL1	PGOL2
Indirizzo:										
Nome:										
Città:										
Oppure:										

Figura 11 - Il nostro terzo Caso Studio. Una query un po' spinta. La Sottoscheda intermedia, che mostra le varie partite disputate, sia in casa che fuori casa, dalla squadra su cui è posizionata la nostra Scheda (nella figura precedente era l'Inter) ha come origine una Query che, a sua volta, usa un parametro passato proprio dalla Scheda superiore (è il campo SNAME). Cambiando la squadra su cui ci si è posizionati la Query viene automaticamente rieseguita e quindi viene riaggiornata la sottoscheda. Da notare l'uso, nella stessa Query, della Function che calcola il segno sulla schedina.

media è quella delle Partite, identificate dalla accoppiata dei nomi delle due squadre che si sono incontrate. Della partita però conviene conservare anche un identificativo (giornata di campionato + progressivo) e altre informazioni, come arbitro, numero di gol, ecc.

Ci sono poi i Giocatori. Pensando ad un solo campionato, sono in relazione Uno a Molti con le Squadre: una Squadra tanti giocatori, un Giocatore una sola Squadra. Dei giocatori memorizziamo il Cognome (che usiamo come campo chiave), il nome della Squadra in cui milita, la Data di Nascita, il Ruolo in cui gioca, identificato da un campo numerico che codifica i ruoli da 1 a 5.

Trascurando il problema delle formazioni, e cioè se quel giocatore ha giocato o meno in quella partita, se l'ha giocata tutta, oppure è entrato, magari come numero 13, oppure se è uscito, magari per espulsione, possiamo mettere i

Giocatori in relazione con le Partite attraverso il concetto di Gol. Anche tra Giocatori e Partite c'è una Matrice rappresentata dal Gol.

Un Gol è identificato dalla partita in cui è stato segnato, dal giocatore che lo ha segnato, dal minuto in cui è stato segnato e, per gli amanti delle codifiche, da un codice che ne indica le modalità di esecuzione (punizione, tiro al volo, gol di testa, autorete, ecc.).

A proposito di codifiche vediamo (in figura 9) come risolvere, con una Function, la codifica che serve per indicare il Ruolo di ciascun Giocatore. Nella Tabella il ruolo è indicato con un numero, che viene scodificato con la funzione RUOLO, nella quale usiamo la istruzione Select Case, più elegante ed importante della Function Choose, che avrebbe in questo caso risolto egualmente il problema. Vediamo anche la Function Segno, che calcola il segno della schedina

sulla base dei due numeri con i gol segnati dalle due squadre.

In un'applicazione, anche di bassa complessità, conviene sempre costruire un Set di Funzioni personalizzate, in aggiunta a quelle proprie del prodotto, che, una volta costruite, possono servire nelle più svariate situazioni.

Chiariti i limiti che ci si dà nel definire l'applicazione, costruite poi le tabelle e le relazioni, si può passare a realizzare le Schede. Ve ne proponiamo una sola (in figura 10) in cui vediamo tre sezioni (scheda con sottoscheda e sottoscheda della sottoscheda). In alto la Squadra, in mezzo una sottoscheda con le Partite giocate dalla squadra, sia in casa che fuori casa. In basso un'altra sottoscheda con i gol segnati nella partita selezionata nella seconda parte sia dalla squadra esaminata che dalla sua avversaria in quella partita.

È interessante esaminare come siano state alimentate le due sottoschede. La prima con una Query che utilizza come parametro il valore della Casella SNAME posto nella parte superiore della scheda e che in tal modo provvede ad estrarre solo le partite giocate dalla squadra in esame. Abbiamo inserito due volte il parametro, una volta sul campo che indica la squadra che gioca in Casa e una volta su quella Fuori Casa. Il segno sulla schedina è calcolato dalla Function (la Query è in figura 11).

Anche la seconda sottoscheda è alimentata da una Query che seleziona solo i gol della partita puntata nella seconda sottoscheda. Per comodità, utilizziamo come campo di passaggio, una casella posta nella sovrascheda in cui visualizziamo la sigla della partita (giornata di campionato + progressivo) selezionata nella sottoscheda di mezzo. Questa casella fa da Campo Master della sottoscheda inferiore, in cui il Campo Figlio è la stessa sigla che contraddistingue i gol segnati nella stessa partita.

## Conclusioni

Abbiamo visto come non siano rari i casi in cui ci si trova di fronte a relazioni Molti a Molti e come sia relativamente semplice risolverli con delle strutture Relazionali standard, facilmente riutilizzabili nelle varie situazioni.

Abbiamo anche potuto constatare come le Schede con Sottoscheda di MS Access siano particolarmente adatte a rappresentare i dati coinvolti in questo tipo di relazione senza dover scrivere una sola riga di programma. E questo anche in quei casi in cui si voglia aumentare il numero delle sottoschede e dei livelli delle sottoschede.

MC