

## Accedere ai database MS SQL Server dai prodotti MS

In casa Microsoft esistono svariate linee di prodotti. Ci sono i prodotti di classe Home, i prodotti di classe Small Office, ambedue destinati a ben individuabili categorie di destinatari, i prodotti Office, destinati agli utenti finali, dai principianti ai cosiddetti "power users", aziendali o meno, i prodotti per chi sviluppa, e quindi i vari linguaggi. Ci sono poi i sistemi operativi, Windows 95 e NT Workstation, per le postazioni client, e Windows NT Server, per i server aziendali. La linea dei prodotti BackOffice, che si affiancano ad NT, comprende numerosi server che espletano i vari servizi necessari in un server tuttofare.

E' chiaro che le varie categorie ed i vari prodotti non possono essere racchiusi in compartimenti stagni, anzi, in certi casi, è proprio la sinergia tra i prodotti delle varie classi che offre la migliore soluzione di un dato problema.

*di Francesco Petroni*

In questo articolo vogliamo approfondire proprio una di queste forme di sinergia, consistente nella possibilità di "vedere" i database, disponibili in formato MS SQL su un server aziendale, direttamente dai vari prodotti client, sia quelli di classe Office che quelli di classe linguaggi.

Da un punto di vista applicativo, una sinergia di questo genere può avere due finalità principali.

La prima, più complessa da realizzare, è quella di costruire applicazioni, aziendaliamente strategiche, in cui i dati risiedono sui server ed i programmi di gestione, almeno per la parte front end, risiedono sui vari client. Per "applicazioni aziendaliamente strategiche" intendiamo applicazioni che "fanno funzionare l'azienda", in cui l'affidabilità del database, l'efficienza dei programmi, la sicurezza del sistema, ecc., siano garantiti.

La seconda finalità è quella di mettere i

dati aziendali a disposizione dei vari utilizzatori, soprattutto per usi manipolativi, e quindi reportistici, statistici, ecc., usi assolutamente non critici in quanto producono, partendo dai database di origine, solo degli output, senza nessun rischio di modificare i dati originali.

Insomma, in un'azienda ci sono da una parte i dati e dall'altra gli utilizzatori. Il problema è quello di mettere a loro disposizione, nella maniera più diretta (senza intermediari), veloce (in termini di tempi di risposta), sicura (senza pericolo di danni) e semplice (modalità operative a portata di un utente finale di competenza media), i dati.

Se i dati sono in formato MS SQL Server e se, dall'altra parte, gli utilizzatori dispongono dei prodotti Office o programmano con il Visual Basic, sono possibili decine di strade alternative per interrogare i database e per ricevere il risultato dell'interrogazione sugli appli-

cativi client.

In questo articolo vedremo qualcuna di queste strade.

### Premesse lato MS SQL

In questo articolo parliamo solo del lato client della faccenda ed eseguiremo dei semplici esperimenti di interrogazione. Chi volesse invece affrontare anche il lato server (e quindi più specificamente MS SQL Server) può leggere la serie di articoli nella rubrica Workgroup, iniziati nel numero scorso e dedicati proprio a MS SQL Server, che sono in un certo senso complementari rispetto a questo. A chi volesse approfondire l'argomento Visual Basic e SQL Server, e volesse soprattutto avere una visione più sistematica delle varie problematiche connesse, consigliamo la lettura dei manuali del Visual Basic (sono ancora fon-

damentali) e del volume Visual Basic & SQL Server di William R. Vaughn, pubblicato nella collana Microsoft Press e ben tradotto in italiano.

Diamo quindi per scontato che sul server ci sia un database di prova (useremo quello che si chiama Pubs, che viene installato quando si installa MS SQL Server). Un secondo database di prova lo installeremo noi, usando la modalità automatica permessa dagli upsizing tool per MS Access.

Per collegarsi al database già disponibile in formato MS SQL, ma il discorso vale per tutti gli altri formati, occorre, sulla postazione client, creare una connessione ODBC, grazie alla quale si assegna un nome al collegamento. Se ci si collega da Excel, Access o Visual Basic, o da qualsiasi altro client, basterà riferirsi a questo nome.

Abbiamo detto che ci occuperemo solo del lato client della faccenda, ma deve essere chiaro che MS SQL Server ha una sua organizzazione da cui non si può prescindere.

Esiste ad esempio il concetto di device, che può essere interpretato come un contenitore "logico" di database. Esiste il concetto di database, che, come appena detto, fa parte di un device e che a sua volta contiene vari tipi di oggetti: table, view, stored procedure, ecc., che possono essere attivati dal client.

Parti essenziali di MS SQL Server sono tutti gli strumenti per la sicurezza degli accessi (l'amministrazione degli utenti, che può essere condivisa con Windows NT), per la sicurezza e dei dati (procedure di backup e di replica), per il rilevamento delle operazioni (si può definire un file di log in cui vengono riportate tutte le operazioni eseguite sui database), per la manutenzione corrente e straordinaria, ed altro.

MS SQL Server deve essere ben configurato non solo dal punto di vista software ma anche dal punto di vista hardware, in modo che anche le prestazioni, in termini di risposta alle interrogazioni, siano ottimizzate. Ad esempio, trattandosi di componenti BackOffice, sia MS SQL Server che Windows NT Server possono essere ospitati su macchine con architettura multiprocessore, con clustering dei dischi rigidi. Un miglioramento delle prestazioni generali del sistema può essere ottenuto "pom-

Figura 1 - MS SQL Server 6.5 - Schematizzazione dei nostri esperimenti.

Scopo di questo articolo è quello di sperimentare come, dai normali prodotti MS Office (useremo Excel, Word e Access) e dai normali linguaggi di programmazione targati Microsoft (useremo Visual Basic) si possa accedere, in varie maniere, ai database in formato MS SQL Server. Le versioni dei vari strumenti che utilizzeremo sono: MS SQL Server 6.5, Office 97, Visual Basic 5.0. Come gestore dei driver ODBC useremo quello presente nel CD di Office (cartella Valuepack\Dataacc).

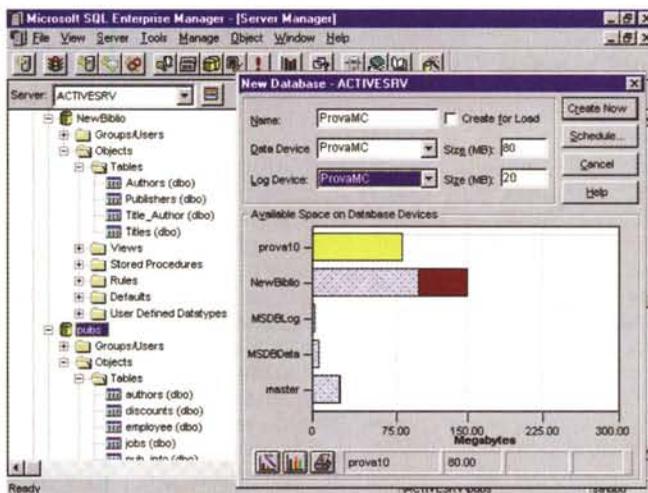
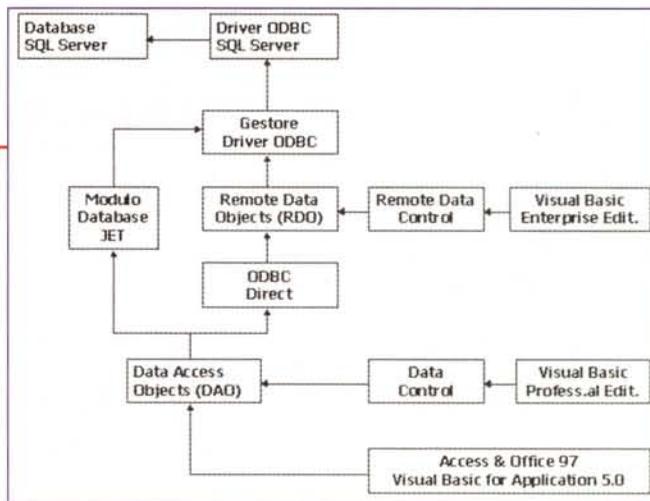


Figura 2 - MS SQL Server 6.5 - Il lato MS SQL della medaglia.

Utilizzeremo da una parte un server Windows NT con installato MS SQL Server 6.5 e dall'altra un client Windows 95 con Office 97 e Visual Basic 5.0. Per i nostri esperimenti useremo due database. Prima di tutto quello che si chiama Pubs e che viene scaricato durante l'installazione dell'SQL Server. Il secondo lo produrremo noi stessi convertendo in formato SQL il database in formato MDB che si chiama Biblio e che è presente nel CD di Visual Basic 5.0. Lo chiameremo NewBiblio. In questa schermata "collage", presa dall'SQL Enterprise Manager, vediamo in secondo piano sia NewBiblio che Pubs, ed in primo piano un momento della creazione di un terzo database, il cui nome è ProvaMC.

pando" il server, piuttosto che intervenendo sui client.

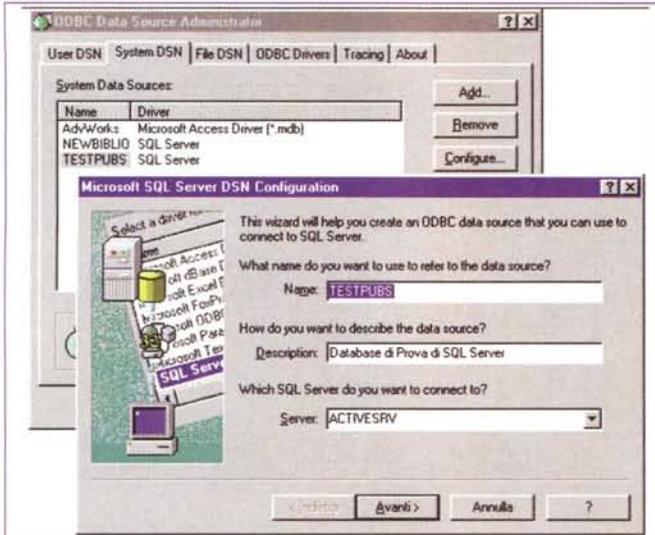
## Figure professionali coinvolte

Ipotizzando un'azienda di dimensioni medie o grandi, in cui i database strategici siano stati realizzati in formato MS SQL Server, occorre prevedere una serie di figure professionali che garantiscano il funzionamento e l'ottimizzazione del sistema.

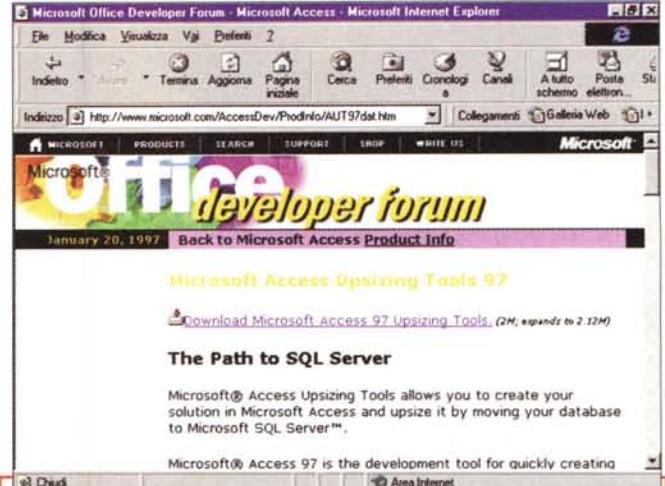
Innanzitutto l'amministratore di MS SQL Server, che potrebbe essere lo stesso che amministra NT, il cui compito

è quello di amministrare gli utenti, di gestire l'organizzazione logica e fisica dei database, di programmare ed eseguire le procedure di replica, di backup, di log, ecc.

Altra figura è quella del progettista del database, il cui compito è quello di costruire la banca dati aziendali sulla base delle finalità che ci si propongono. Si tratta di un'attività di analisi sui dati sulle procedure aziendali che si concretizza in un progetto di database ed eventual-



**Figura 3 - MS SQL Server 6.5 - Configurazione della connessione ODBC.**  
I rapporti tra client, ovvero tra le applicazioni presenti sulla postazione client, e server, ovvero i due database, in formato SQL, presenti sul server NT, sono intermediati dal driver ODBC, che va opportunamente configurato sulla postazione client. Le varie impostazioni della connessione ODBC valgono per qualsiasi tipo di banca dati si voglia raggiungere. Nel nostro caso abbiamo creato la connessione TestPubs che accede al database Pubs disponibile sul server, che a sua volta si chiama ACTIVESRV. In un secondo momento creeremo la connessione NewBiblio, verso il database NewBiblio. Qualche cosa in più, su concetto di driver ODBC, la diciamo nel testo.



**Figura 4 - MS Access 97 upsizing tool -** Come al solito molto del materiale si trova su Internet.  
Dal sito Microsoft potete scaricare molto materiale utile per i vostri esperimenti. Innanzitutto Microsoft Access 97 upsizing tool, che vedremo subito dopo. Si può scaricare anche una nuova versione, per chi avesse la vecchia o per chi non l'avesse affatto, del file Biblio.mdb. Altro sito da tenere sotto controllo è [www.microsoft.com/data](http://www.microsoft.com/data) nel quale potete trovare materiale illustrativo delle nuove strategie di accesso ai dati e parecchio materiale d'esempio. Citiamo quello relativo alla nuova tecnologia UDA (Universal Data Access) grazie alla quale le varie tecniche di accesso ai dati vengono, in un certo senso, unificate, soprattutto in previsione della diffusione dell'accesso ai dati attraverso Internet.

mente nella sua realizzazione. E' necessaria una competenza da analista ed una conoscenza di MS SQL Server in quanto è con quest'ultimo che saranno realizzate tabelle, procedure, ecc. Nel caso in cui si persegua la filosofia del data warehouse (magazzino dei dati aziendali), è necessario prevedere uno specialista SQL a supporto delle richieste degli utenti. Se ad un utente servono, per svolgere la propria attività, sottoinsiemi di dati provenienti dai database aziendali, è bene che questi dati siano preparati da una procedura, scritta nel linguaggio specifico di MS SQL Server, che viene eseguita dal server stesso periodicamente o a richiesta. In pratica questo esperto realizza una stored procedure, memorizzata all'interno di MS SQL Server, che viene richiamata dal client dell'utente ma viene eseguita dal server. Ad utente particolarmente esperto si potrebbe concedere una certa libertà d'azione, nel senso che gli sarà consentito di realizzare in proprio le interrogazioni utilizzando strumentazioni client. Ultima figura da prevedere, nel caso si realizzino procedure evolute, è quella del programmatore, figura che potrebbe coincidere, ma non obbligatoriamente, con il progettista del database. Co-

stui deve conoscere il linguaggio che si decide di usare per realizzare l'applicazione (ad esempio MS Visual Basic) e il database che ospita i dati (ad esempio MS SQL Server). Le più recenti tecniche di programmazione prevedono lo sfruttamento di un'architettura client/server a tre livelli, in cui, sostanzialmente, anche buona parte dei programmi, escluse le sole procedure che gestiscono il front end, risiedono sul server. Ma questo è un argomento fuori tema, che tratteremo un'altra volta.

## Elenco sintetico dei nostri esperimenti

Condurremo i nostri esperimenti utilizzando un server NT 4.0, con installato MS SQL Server 6.5 (figura 1). Lavoreremo sul database di prova Pubs e su un altro database di prova che costruiremo dal nulla. Daremo un'occhiata, più che altro per ambientarci, ad esempio per individuare le tabelle, per "vedere come è fatta" una stored procedure, ecc., a MS SQL Server (figura 2 e figura 14). Poi ci trasferiremo sul client, nel quale creeremo una connessione ODBC (fi-

gura 3) che proveremo subito per accedere al database Pubs, da Visual Basic, da Access, da MS Query (e quindi da Excel), da Word. A questo punto faremo un intervallo dedicato allo studio degli Access upsizing tool, scaricabili gratuitamente da Internet (figura 4). Si tratta di due strumenti molto comodi quando si debbono far dialogare database SQL e postazioni client. Il primo dei due è un browser che, lanciato da MS Access, permette di navigare all'interno del database SQL (figura 5), il secondo è un wizard che permette di creare, partendo da un database in formato Access (desinenza MDB), direttamente un database SQL. Per provare questa procedura partiremo dal database Biblio.mdb, in dotazione a MS Visual Basic 5.0, di cui vediamo in figura 6 la struttura in termini di tabelle e relazioni. C'è da aprire una piccola parentesi dedicata a chi dispone di Access ma non di Visual Basic. Sia gli upsizing tool che Biblio.mdb (in una versione più aggiornata, che è quella che abbiamo utilizzato anche noi) sono scaricabili dal sito Microsoft. Questo permette di eseguire buona parte dei nostri esperimenti anche a chi non disponga di Visual Basic.

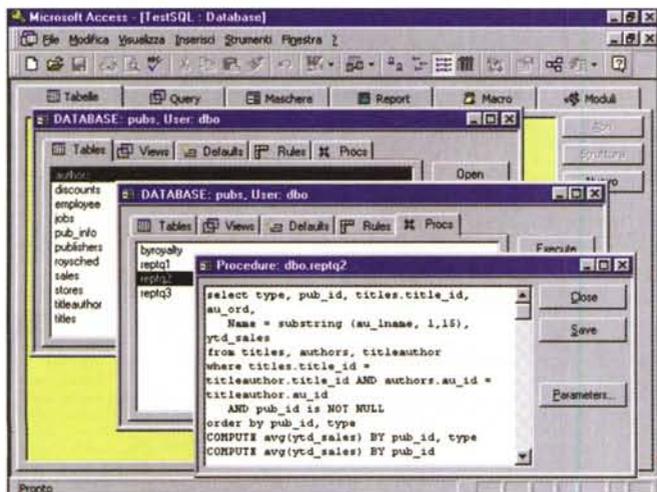


Figura 5 - MS Access 97 upsizing tool - Cominciamo con il browser. Scaricati gli upsizing tool ed installati sulla postazione client possiamo avviarli dal menu Access passando per la voce strumenti aggiunti. La prima delle due procedure che vengono aggiunte è quella che permette di navigare all'interno di una connessione ODBC. In questa figura vediamo un collage di tre box create dal browser. La prima mostra il contenuto, in termini di table, view, default, rule, proc (stored procedure) del database Pubs, ed in particolare stiamo vedendo la lista delle tabelle. Nella seconda vediamo le proc e nella terza la sintassi di una specifica procedura, che si può modificare, anche inserendovi dei parametri. Insomma si riesce ad intervenire, lavorando esclusivamente in Access, non solo sui dati, ma anche sui componenti del database SQL.

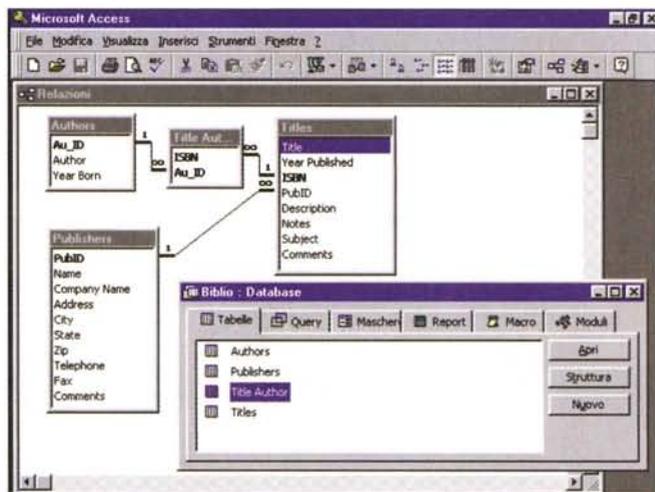


Figura 6 - Biblio.mdb - Il nostro database di prova. Il secondo tool degli upsizing tool è quello che converte un normale database, già disponibile e pieno di dati in formato Access, in un database in formato MS SQL Server. Come database di partenza usiamo quello che si chiama Biblio.mdb ed è presente nel CD del Visual Basic oppure è scaricabile, in un'edizione più aggiornata, dal sito della Microsoft dedicato al Visual Basic. Riguarda la problematica biblioteca, in cui ci sono le tabelle degli editori (publisher), degli autori (author), dei titoli (title) e per risolvere il rapporto tra titoli e autori (un titolo più autori, un autore più titoli) c'è una quarta tabella title\_author. I titoli sono veri e sono oltre 10.000. Quindi gli esperimenti risulteranno significativi anche in termini di volumi.

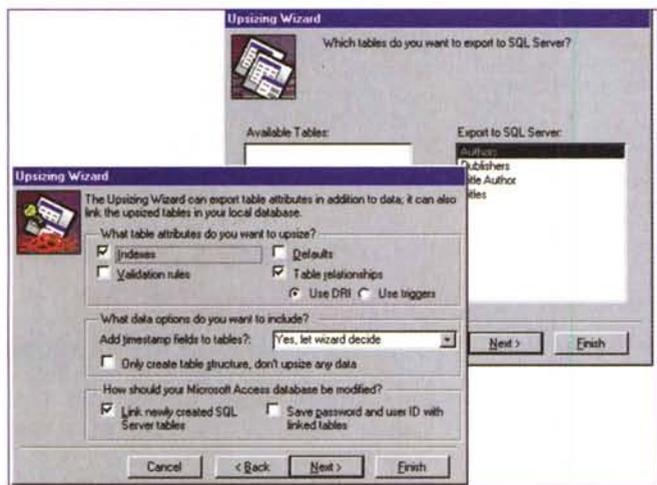


Figura 7 - MS Access 97 upsizing tool - Due vedute del wizard che guida l'upsizing. E' chiaro che per un efficace utilizzo di tale strumento occorrerebbe una buona conoscenza sia dell'ambiente SQL Server che dell'ambiente Access, in modo tale che i vari elementi presenti nel secondo siano correttamente convertiti in elementi del primo. Nella peggiore delle ipotesi si può convertire solo la struttura delle tabelle ed il contenuto delle stesse, ignorando le regole di validità, le regole relazionali, gli indici, e quanto altro, impostato in Access, non si sa però come convertire in SQL. Qui vediamo due momenti dell'"interrogatorio" eseguito dalla procedura di auto-composizione. Chiameremo la connessione NewBiblio così come il nuovo database.

Installati gli upsizing tool e caricato Biblio.mdb possiamo lanciare il wizard che crea il nostro nuovo database. In figura 7 ne vediamo due momenti. Tanto per capirci il server si chiama ACTIVESRV, il database lo abbiamo chiamato NewBiblio e lo stesso nome lo abbiamo affibbiato alla connessione.

Il modo più semplice per provare la nuova connessione è quello di eseguire il comando Dati - Carica dati esterni - Crea nuova query di Excel, che lancia l'accessorio MS Query (in figura 8), che permette di costruire in un ambiente guidato, di tipo QbE (Query by Example), la nostra interrogazione, il

cui risultato viene scaricato direttamente sul foglio di Excel. Nella figura 9 vediamo proprio il foglio Excel ed, in primo piano, la box dalla quale si deve "passare" per accedere al nostro database SQL.

Indipendentemente dall'uso e dalla presenza degli upsizing tool, se si vogliono utilizzare da Access i dati dal database SQL sono praticabili varie strade.

La più diretta è quella di creare delle tabelle "attached" con il comando (siamo in Access) File - Carica dati esterni - Collega tabelle. A questo punto le tabelle sono utilizzabili come se fossero tabelle di Access a tutti gli effetti. Non sono ovviamente possibili modifiche a livello di struttura. Altra possibilità è quella creare delle query di tipo specifico (comando Query - SQL specifico - Pass-through). Occorre scrivere a mano (non si può usare il QbE) un'istruzione SQL nello specifico linguaggio SQL del server. Tale istruzione salta il motore Jet di Access e viene eseguita direttamente dal server. Se poi fossero disponibili delle stored procedure, si possono eseguire, dall'interno della query pass-through, con l'istruzione Exec nomeprocedura. Tutte queste cose appena dette per Access le abbiamo riassunte nel collage di figura 10.

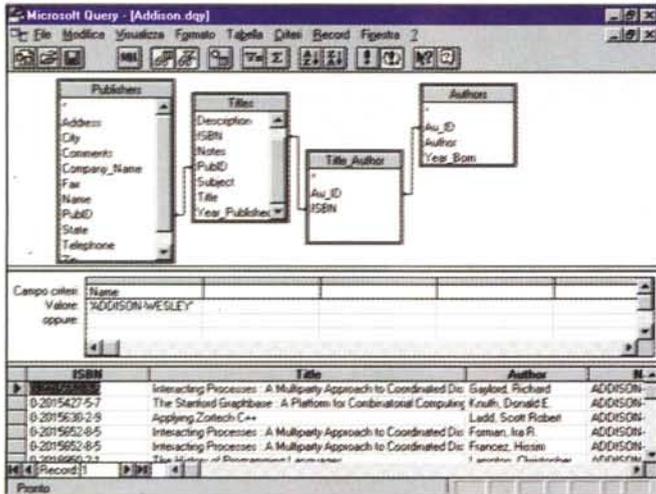


Figura 8 - MS Query 97 - Lo strumento generico di interrogazione di MS Office. Qui vediamo in azione lo strumento MS Query, che viene installato quando si installa Office 97, e che può essere richiamato direttamente da Excel con comando Dati - Carica dati esterni - Crea nuova query. L'ambiente operativo, del tutto analogo all'ambiente query di Access, permette in maniera guidata di creare una interrogazione sui nostri due database, raggiunti tramite le connessioni TestPubs o NewBiblio.

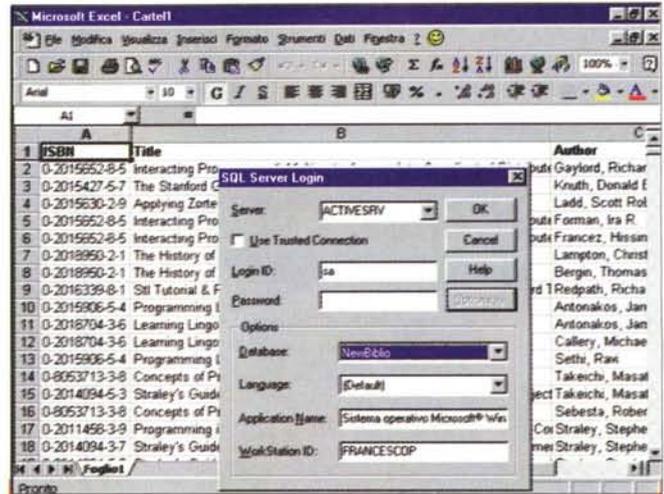


Figura 9 - MS Query 97 - A servizio di Excel. Qui vediamo, in un collage, sia il nostro foglio Excel riempito con i dati provenienti dall'MS Query sia la box che viene visualizzata quando si accede alla connessione NewBiblio. Una volta creata la query e scaricati i dati sul foglio sono attivabili, ad esempio attivando il quick menu sull'area con i dati, i comandi di Aggiornamento dei dati oppure di Modifica della query. In altre parole l'area con i dati rimane legata all'origine stessa dei dati. L'intera operazione di query si può parametrizzare e si può inglobare in una macro. In questo modo si riesce a creare una procedura generalizzata di interrogazione del database SQL, eseguibile totalmente da Excel.

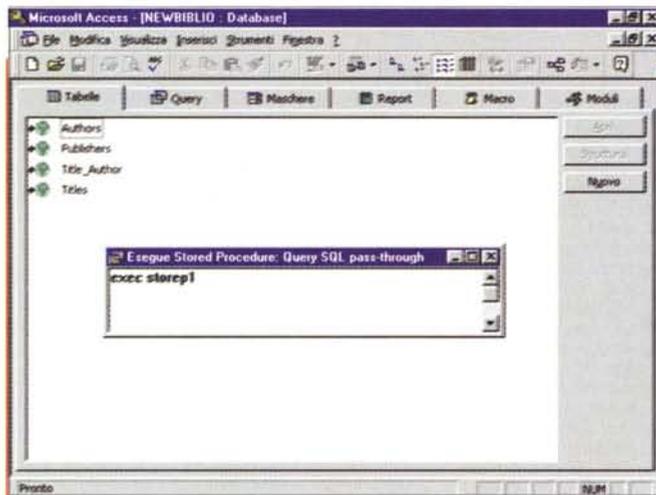


Figura 10 - MS Access 97 - Tabelle "attached" e query specifiche. Verificato come Excel possa approfittare delle ottime capacità di MS Query anche per interrogare database SQL, passiamo ad Access. La prima cosa che vediamo è come si possa costruire un'applicazione client che usi le tabelle messe a disposizione dal nostro database Pubs, in formato SQL, raggiunto tramite la connessione TestPubs. Con il comando File - Carica dati esterni - Collega tabelle si creano delle tabelle "attached", che

possono essere utilizzate per alimentare maschere o report, ecc., come se fossero normali tabelle Access. Se invece occorre pretrattare i dati, creando ad esempio delle tabelle virtuali ottenute da query eseguite sul database di origine, la strada migliore è quella di creare, nell'ambiente SQL Server, una stored procedure e di richiamarla dall'ambiente client, come query pass-through. La sintassi è banale.

base SQL Server, il remote data control (lo vediamo nelle figure 11 e 12), oggetto specializzato, disponibile solo nella versione Enterprise di Visual Basic 5.0. Il secondo invece sfrutta la programmazione RDO (lo vediamo in figura 13 e 15, che mostra il listato in edizione integrale). Occorre caricare la libreria

MS Remote Data Object (il cui file è MSRDO20.DLL), in modo che siano inseribili, nel codice VB, anche le istruzioni RDO. La programmazione RDO, così come quella DAO, è praticabile anche da Visual Basic for Application versione 5.0 e quindi anche in tutti i prodotti Office compatibili.

## Creazione ed uso della connessione ODBC

ODBC è una delle voci richiamabili dal pannello di controllo di Windows 95. Permette di creare dei system data source ODBC, ovvero delle connessioni a fonti di dati, identificabili con un nome, imposto da noi a livello di client (lo chiamiamo TestPubs, che fa riferimento ad una macchina server, nel nostro caso il server si chiama ACTIVE-SRV, e ad un database in essa disponibile, nel nostro caso usiamo Pubs, installato con MS SQL Server). Eseguite queste semplici impostazioni diventa possibile riferirsi, da qualsiasi applicativo client, alla connessione stessa per reperire i dati. E' chiaro che non basta conoscere il nome della connessione, ma occorre anche sapere come è fatto il database, ovvero quali sono le tabelle ed i campi.

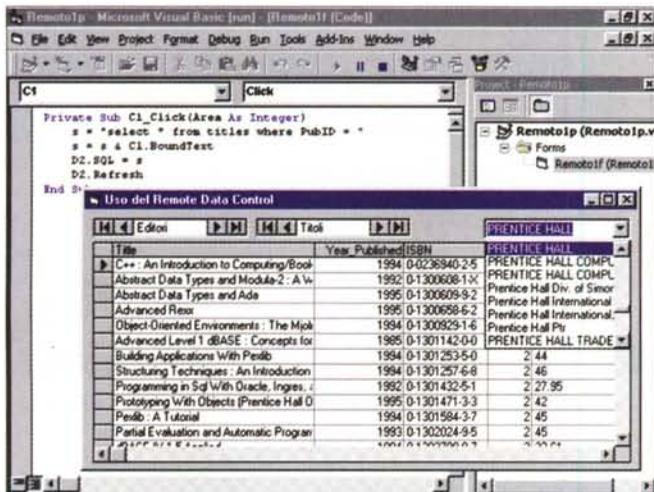


Figura 11 - MS Visual Basic 5.0 - Enterprise Edition - Utilizzo del remote data control - Esecuzione.

Anche partendo da un'applicazione Visual Basic esistono numerose strade alternative per raggiungere dati disponibili in formato SQL Server. Si può usare il normale data control, passandogli tre o quattro proprietà, oppure la programmazione degli oggetti DAO. Se si dispone della versione Enterprise di VB si può usare il remote data control, oppure si può usare la programmazione degli oggetti RDO (remote data object). Queste strade sono abbastanza simili alle prime due, solo che sono specializzate per lavorare sui dati disponibili sul server. Esistono altre strade, più impegnative, ad esempio quelle che accedono direttamente alle DLL per SQL, ma che non vedremo in questa sede. Nel primo esperimento useremo il remote data control, collegato alla connessione NewBiblio.

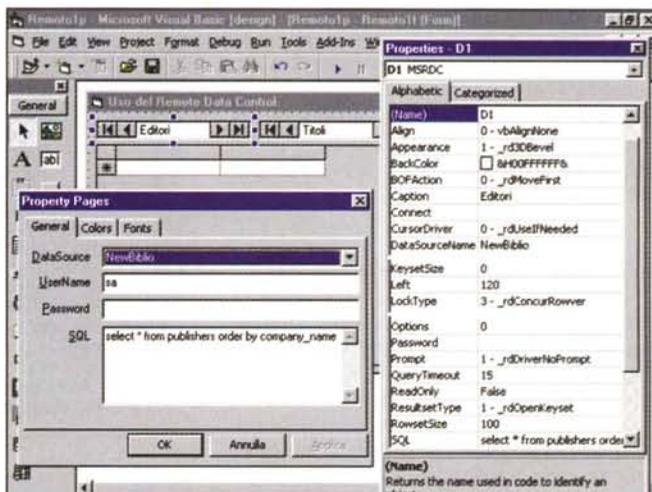
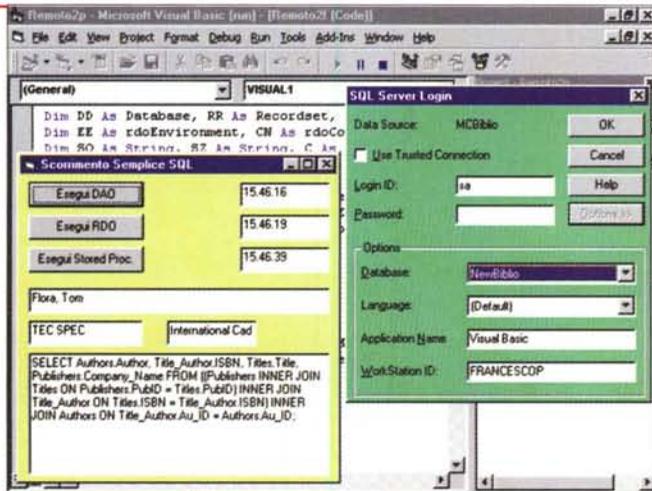


Figura 12 - MS Visual Basic 5.0 - Enterprise Edition - Utilizzo del remote data control - Impostazione delle proprietà.

L'obiettivo è quello di elencare i titoli del nostro database NewBiblio, selezionandoli per editore. Usiamo due remote data control (RDC), il primo legato alla tabella publisher (editori) ed il secondo legato ad una query da generare dinamicamente. Al primo RDC colleghiamo una DBCCombo (proprietà RowSource della DBCCombo) che visualizza i nomi degli editori (proprietà ListField, campo Company\_name) ma che assume il valore (proprietà BoundColumn, campo PubID) del suo codice. Al click sulla DBCCombo, quindi dinamicamente, generiamo un'istruzione SQL che seleziona tutti i titoli di quell'editore. La nuova istruzione la passiamo al secondo RDC che a sua volta alimenta una DBGrid. Si tratta di applicazioni tipiche di Visual Basic che un utilizzatore "medio" deve saper realizzare in cinque - dieci minuti.

Figura 13 - MS Visual Basic 5.0 - Enterprise Edition - Utilizzo della programmazione degli oggetti DAO e RDO - Esecuzione.

Il secondo programma realizzato con Visual Basic mette a confronto tre metodi di programmazione. Lo scopo, banale, è quello di eseguire una query che "pesca" dati dalle quattro tabelle del database NewBiblio e li scorre dal primo all'ultimo (sono quasi 13.000 record). La prima volta usiamo la programmazione DAO, la seconda la programmazione RDO, la terza invece lanceremo una stored procedure creata in SQL. Eseguiremo anche il cronometrando delle prestazioni, da valutare cambiando i vari parametri di configurazione, che, soprattutto nella programmazione RDO, alterano pesantemente i tempi di risposta.



## MS Access upsizing tool

Abbiamo visto come ci si collega ad un database SQL già disponibile (Pubs). Vediamo ora come, direttamente dalla postazione client, sia possibile promuovere in "Serie A", ovvero convertire in SQL Server, un database già realizzato con Access. Occorre utilizzare l'utility di cui abbiamo parlato prima, scaricabile da

Internet (dove e come lo vedete in figura 4) e che si installa come aggiuntivo di Access 97, all'interno del quale inserisce due nuove voci menu:

- Strumenti Aggiuntivi Browser
- Strumenti Aggiuntivi Upsize to SQL Server

La prima permette, noto il nome della connessione, di navigare all'interno del database e di vedere le table, le view, le rule e le procedure. Una volta analizzati gli elementi ci si può collegare alle tabelle (tabelle attached) e si possono eseguire delle query di tipo pass-through, che eseguono direttamente le stored procedure residenti sul server.

Tutt'altra finalità ha il wizard che esegue il processo di upsize. In pratica produce un database SQL partendo da un "modellino" fatto con Access. Ho usato la parola modellino per indicare quello che è il vero scopo degli upsizing tool. Non tanto quello di evitare di costruire le strutture dati lavorando con Access anziché con SQL Server, quanto quello di consentire di lavorare per prototipi anche in applicazioni aziendali importanti. Costruire un'applicazione locale con Access è almeno venti volte più economico del costruirla direttamente con SQL. Con il prototipo è già possibile eseguire tutti i test operativi, necessari a convalidare la correttezza del database e il funziona-

mento delle sue procedure.

Una volta a punto il prototipo lo si può convertire, a costi bassissimi, in un'applicazione aziendale.

Per quanto ci riguarda abbiamo eseguito l'upsizing del database Biblio.mdb, ed abbiamo chiamato NewBiblio la connessione.

## Un po' di esperimenti

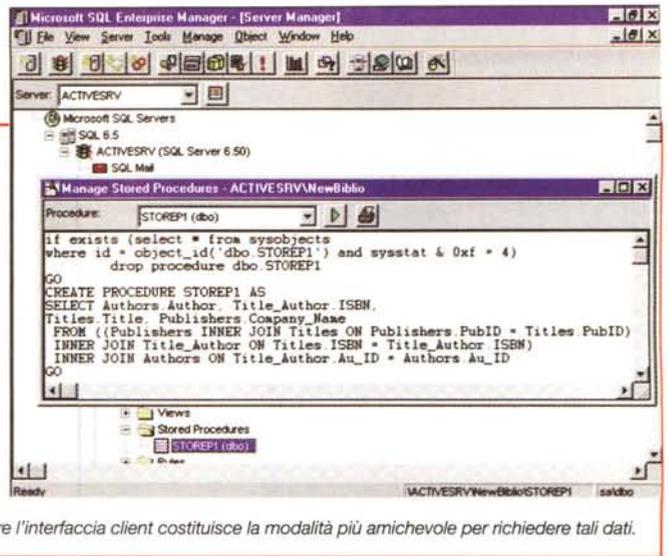
Abbiamo parlato di SQL Server, di connessioni ODBC, di upsizing tool ed abbiamo brevemente descritto gli esercizi proposti. Pensiamo che, con queste brevi descrizioni e con l'aiuto delle figure qui intorno, siate in grado di eseguirli anche voi e sugli stessi nostri dati che trovate in MS SQL Server ed in Visual Basic.

Abbiamo lavorato con MS Query e con Excel, per creare delle interrogazioni il cui risultato viene riversato sul normale foglio di lavoro per le ulteriori elaborazioni.

Abbiamo lavorato con Access, per collegarci a tabelle esterne SQL, per eseguire delle stored procedure che per Access sono delle query analoghe alle altre. In pratica sia le tabelle "attached" che le query pass-through costituiscono delle tabelle virtuali e possono essere utilizzate per creare maschere, per creare report.

Figura 14 - MS SQL Server 6.5 - La nostra stored procedure Storep1.

Vediamo come è fatta la stored procedure costruita direttamente in MS SQL Server e richiamata dalla nostra applicazione VB. E' evidente che un'applicazione client/server è ottimizzata quando ognuno dei due componenti, il server ed il client, dà il meglio di sé. La stored procedure confezionata dal programmatore SQL costituisce sicuramente la strada più sicura e veloce di accesso ai dati, mentre l'interfaccia client costituisce la modalità più amichevole per richiedere tali dati.



```
Dim DD As Database, RR As Recordset, QD As QueryDef
Dim EE As rdoEnvironment, CN As rdoConnection, RS As rdoResultset
Dim SQ As String, SZ As String, C As Long

Private Sub Form_Load()
    SQ = "SELECT Authors.Author, Title_Author.ISBN, Titles.Title, Publishers.Company_Name"
    SQ = SQ + " FROM ((Publishers INNER JOIN Titles ON Publishers.PubID = Titles.PubID)"
    SQ = SQ + " INNER JOIN Title_Author ON Titles.ISBN = Title_Author.ISBN)"
    SQ = SQ + " INNER JOIN Authors ON Title_Author.Au_ID = Authors.Au_ID;"
    T0 = SQ
End Sub

Private Sub P1_Click()
    CC = "ODBC; DSN=NewBiblio; UID=sa"
    Call Via: Set DD = OpenDatabase("NewBiblio", False, False, CC)
    Set RR = DD.OpenRecordset(SQ, dbOpenSnapshot)
    Call Inizio: Do While Not RR.EOF: ` VISUAL1
        C = C + 1: RR.MoveNext
    Loop: Call Fine
End Sub

Private Sub P2_Click()
    strConnect = "": Call Via
    Set En = rdoEnvironments(0)
    En.CursorDriver = rdUseIfNeeded
    Set CN = En.OpenConnection_
        (dsName:="NewBiblio", Prompt:=rdDriverNoPrompt, ReadOnly:=False, Connect:=strConnect)
    Set RS = CN.OpenResultset(Name:=SQ, Type:=rdOpenDynamic, LockType:=rdConcurValues)
    Call Inizio: Do While Not RS.EOF: ` VISUAL2
        C = C + 1: RS.MoveNext
    Loop: Call Fine
End Sub

Private Sub p3_Click()
    strConnect = "": SP = "exec storep1": Call Via
    Set En = rdoEnvironments(0)
    En.CursorDriver = rdUseIfNeeded
    Set CN = En.OpenConnection_
        (dsName:="NewBiblio", Prompt:=rdDriverNoPrompt, ReadOnly:=False, Connect:=strConnect)
    Set RS = CN.OpenResultset(Name:=SP, Type:=rdOpenDynamic, LockType:=rdConcurValues)
    Call Inizio: Do While Not RS.EOF: ` VISUAL2
        C = C + 1: RS.MoveNext
    Loop: Call Fine
End Sub

Sub VISUAL1()
    T1 = RR!Author: T2 = RR!Title: T3 = RR!Company_Name: DoEvents
End Sub

Sub VISUAL2()
    T1 = RS!Author: T2 = RS!Title: T3 = RS!Company_Name: DoEvents
End Sub

Sub Via()
    M0 = Time: M1 = "": M2 = "": DoEvents: C = 0: Screen.MousePointer = vbHourglass
End Sub

Sub Inizio()
    M1 = Time: DoEvents: C = 0: Screen.MousePointer = vbHourglass
End Sub

Sub Fine()
    Screen.MousePointer = vbDefault: M2 = Time: DoEvents: MsgBox C
End Sub
```

Figura 15 - MS Visual Basic 5.0 - Enterprise Edition - Utilizzo della programmazione degli oggetti DAO e RDO - Listato.

Nel listato notiamo: le dichiarazioni iniziali, la lunga istruzione SQL che esegue un join sulle quattro tabelle del nostro database, le tre routine che stiamo provando, la prima DAO, la seconda RDO e la terza che esegue sul server, lanciandola dal client, una stored procedure creata con SQL Server. Le ultime routine sono quelle che calcolano i tempi di esecuzione e li visualizzano sulla form.

## MS Visual Basic, liscio e for Application, e MS SQL Server

Prima di parlare dei nostri due esercizi, realizzati con Visual Basic, dobbiamo fare una premessa.

Visual Basic non ha un suo specifico formato dati. La Microsoft, casa produttrice del Visual Basic, ne suggerisce due: il formato MS Access, per le applicazioni monoutente o per le applicazioni che lavorano su reti medio-piccole, e il formato MS SQL, per le applicazioni destinate a lavorare su grossi volumi di dati e su reti medio-grandi.

Nel "pacchetto" Visual Basic 5.0 è presente anche una versione "per sviluppatori" dell'SQL, che consente comunque di completare e di testare l'applicazione, prima ancora di aver acquistato MS SQL Server.

Lavorare con database in formato SQL vuol dire cominciare a pensare in termini client/server.

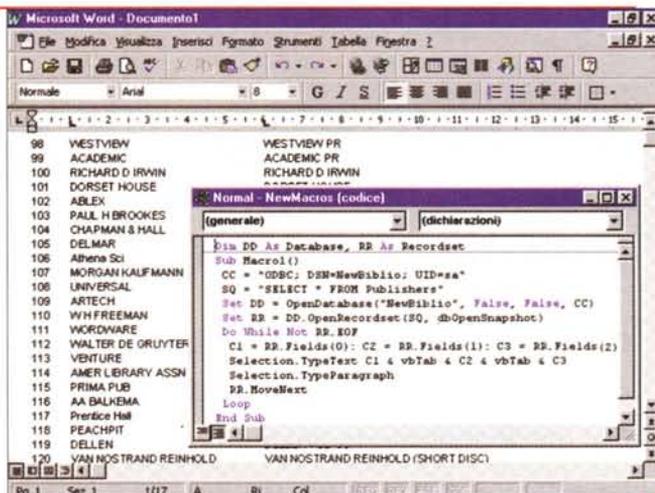
Per Microsoft un'applicazione client/server è un'applicazione che ha i dati sul server, i motori di accesso ai dati sul server, i componenti principali dell'applicazione anche questi sul server, e solo i componenti front end, quelli che costituiscono l'interfaccia che vede fisicamente l'utente, sulle postazioni client.

Non vogliamo, in questa sede, parlare di client/server, ci limitiamo a mostrare un paio di esercizi, realizzati con Visual Basic, Enterprise Edition, che sfruttano componenti e librerie orientati al server e quindi sostanzialmente adatti ad essere usati con database SQL.

Sperimentiamo solo due modalità tra le

Figura 16 - MS Word 97 - La prova del 9.

Tutti i metodi di accesso ai dati praticabili con Visual Basic 5.0, sono, per definizione, utilizzabili da tutti gli applicativi che utilizzano come linguaggio macro il VBA 5.0, primi tra tutti i componenti di Office 97. Qui vediamo un elenco ricavato dalla tabella publisher del nostro database di prova NewBiblio e realizzato con una macro di Word (in primo piano) che accede ai dati direttamente, utilizzando la tecnologia DAO. Se vediamo il problema dal punto di vista Word, è come se Word fosse un generatore di report di dati presenti in un qualsiasi database. E' necessario, perché il programma possa funzionare, referenziare la libreria DAO 3.5, che viene installata sia da VB 5.0 che da Office 97.



però utilizzano il motore Jet, al contrario di RDC e RDO.

Non parliamo neanche delle altre modalità, ancora più sofisticate, che sfruttano componenti che si trovano solo nei Programmer's Toolkit, come le Visual Basic Library for SQL Server.

Cerchiamo di capire come funziona il primo esercizio guardando la figura 11.

L'utente sceglie da una DBCombo un editore (es. PRENTICE HALL) e una DBGrid viene riempita con i titoli pubblicati da quell'editore.

Utilizziamo due controlli RDC (che vediamo in alto ma che potrebbero essere impostati come non visibili). Il primo, legato alla tabella publisher, alimenta una DBCombo per mostrare in essa l'elenco degli editori. Elenchiamo le proprietà fondamentali dei due primi oggetti:

Remote Data Control	Name	D1
	DataSourceName	NewBiblio
	SQL	Select * from publishers order by company_name
DBCombo	Name	C1
	RowSource	D1
	ListField	Company_name
	BoundColumn	PubID

tantissime possibili, la prima che sfrutta il remote data control (RDC) e la seconda che sfrutta la programmazione degli oggetti RDO (remote data object). Queste due modalità sfruttano componenti (il controllo RDC e la libreria RDO) presenti solo nella Enterprise Edition del Visual Basic 5.0.

Con la versione Professional (che non ha né RDC né RDO) si può ricorrere al classico data control oppure alla programmazione degli oggetti DAO, che

I secondi due oggetti sono un nuovo RDC e una DBGrid:

Remote Data Control	Name	D2
	DataSourceName	NewBiblio
	SQL	impostato dinamicamente
DBGrid	Name	G1
	RowSource	D2

Se l'utente seleziona un editore nella DBCombo, viene visualizzato il nome dell'editore e la sua proprietà Bound-

Text assume il valore del suo codice che è quello che serve per la selezione dei titoli.

A questo punto gestiamo l'evento click su C1, con il quale confezioniamo una nuova istruzione SQL che passiamo dinamicamente al secondo remote data control. Il listato, minimo, si può leggere in figura 11.

Il secondo esercizio VB5 (figure 13 e 15) sfrutta sia il codice DAO che il codice RDO. Mette a confronto tre metodi di accesso a programma: DAO, RDO e il richiamo di una stored procedure, che vediamo in figura 14. Vi proponiamo il listato completo e un po' di commento nelle didascalie.

## Ultimissima ed importantissima cosa

Visual Basic 5.0 e Visual Basic for Application 5.0 sono del tutto allineati in termini di... programmabilità. Questo allineamento riguarda anche le routine di accesso ai database, che se scritte bene per il VB funzionano sicuramente anche in VBA 5.0. Nell'ultima figura vediamo una macro VBA scritta per Word 97, che produce un documento con l'elen-

co dei publisher, leggendo, con la tecnologia DAO, direttamente il nostro database SQL.