

Cenni di programmazione

MS Access per chi sa il dBase, ma anche per chi non lo sa

Quarta ed ultima puntata

Si tratta dell'ultimo di una serie di quattro articoli di approfondimento scritti successivamente alla prova di Microsoft Access.

Dopo aver parlato di Tabelle e di Query, nel primo articolo, di Schede, nel secondo, di Report, nel terzo e precedente articolo, ora affrontiamo l'argomento Programmazione, che coinvolge due ulteriori «oggetti» di Access: le Macro e le Procedure

di Francesco Petroni

L'argomento è complesso, per cui ci siamo premuniti, con quella parolina «cenni» messa in alto. In realtà più che descrivere il linguaggio di programmazione vogliamo descrivere i meccanismi attraverso i quali una applicazione Access sviluppata, oggetto per oggetto e interattivamente, può diventare una Applicazione «chiusa».

Alla fine cercheremo di tirare le fila del discorso, confrontando Access con i potenziali concorrenti, che sono i vari Paradox per Windows, dBase IV versione 2, della Borland, i vari Clipper e dBase della C.A., DataEase della DataEase, e FoxPro, biplatforma (per DOS e Windows) della Microsoft stessa.

Breve riassunto delle puntate precedenti

Nella prima puntata abbiamo presentato il nostro Caso Studio, basato su tre Tabelle, ricadenti in una problematica Stipendi, e poste in relazione tra di loro.

C'è la Tabella con i Dati Anagrafici dei Dipendenti (DATIUNO), che è in relazione Uno a Uno con la seconda Tabella Dati Mensili (DATIDUE). I campi su cui è basata la relazione si chiamano COD in tutte e due le Tabelle.

Ogni dipendente appartiene ad una Qualifica e su questa è basata la terza Tabella (DATITRE) che contiene i dati

numerici di base che servono per il calcolo degli Stipendi e che sono organizzati proprio per qualifica.

La tabella delle Qualifiche è in relazione Uno a Molti con quella Anagrafica, in quanto ad una Qualifica corrispondono molti Dipendenti (al campo CQUAL di Datitre corrisponde il campo Q di Datiuno).

Nella prima puntata abbiamo parlato di Tabelle e di Query, e abbiamo anche sottolineato l'importanza di questo secondo strumento che, lavorando con dei «Dynaset» di dati, permette viste dirette sui dati, che sono quindi aggiornabili dalla Query stessa, limitatamente a quanto permesso dalle regole relazionali.

Nella seconda puntata abbiamo parlato di Schede e nella terza di Report. Si tratta di due oggetti abbastanza simili, che si possono appoggiare, nel senso che ne visualizzano i dati e ne permettono la manipolazione, su Tabelle e su Query.

Ne abbiamo realizzati un bel po', descrivendo sia l'ambiente operativo, nel quale si disegnano letteralmente Schede e Report, sia le possibilità in termini di definizione di campi (anche alla Windows), di controllo e di calcolo dei dati immessi, ecc.

In questo numero parliamo dei due ultimi Oggetti, le Macro e i Moduli, che sono i due oggetti su cui si basa la programmabilità di una applicazione Access.

Macro e Moduli

Gli oggetti che compongono un'applicazione Access sono di sei tipi. Tabelle, Query, Schede e Report sono gli oggetti interattivi, creabili ed utilizzabili direttamente anche dall'utente meno esperto.

Gli ultimi due oggetti, Macro e Moduli, rappresentano i due livelli possibili di programmazione di Access.

La Macro consiste, come in tutti gli altri prodotti di Personal Computing, nella memorizzazione di una serie di Azioni, e nella possibilità della loro diretta ed immediata esecuzione con un unico comando.

La Azioni inserite in una Macro possono essere Azioni eseguibili normalmente usando interattivamente Access (es. Apri Scheda), oppure Azioni utili in una Macro e non eseguibili interattivamente (es. apparizione di una Message Box).

La Macro può essere eseguita direttamente dalla Finestra Database, selezionandola e pigiando il bottone Esegui, oppure, più elegantemente, può essere collegata ad un Bottone personalizzato inserito in una Scheda, al limite in una Scheda creata solo per contenere Bottoni.

I Moduli costituiscono il livello più evoluto della programmazione, vanno predisposti sfruttando uno specifico Editor e usando il linguaggio Access Basic, assolutamente analogo al Visual Basic.

I Moduli possono contenere Funzioni oppure Procedure.

La differenza principale tra i due «soggetti» sta nel fatto che una Funzione restituisce un risultato, mentre una Procedura non lo può fare.

Una Funzione può essere molto complessa, in quanto accetta (al contrario di quanto ad esempio permette la Macro Funzione dell'Excel) istruzioni Logiche, istruzioni di Salto e Istruzioni Cicliche.

Alla scoperta delle caratteristiche delle Macro e dei Moduli

Per scoprire le problematiche legate ai vari oggetti e le loro modalità di utilizzo andiamo a curiosare nell'applicazione GESTORD, che fa parte della dotazione di Access, e che sfrutta tutte le possibilità in termini di automazione e di programmazione. Per questo motivo rappresenta una vera e propria miniera di informazioni.

Al lancio dell'applicazione appare una Scheda, che fa da menu. Pigiando ognuno dei quattro Bottoni Grafici, nella Box, in mezzo alla scheda, appare una diversa Lista di Voci di Menu. Evidenziando una voce e pigiando il bottone Selezione si lancia il sottoprogramma, costituito da una Macro.

In figura 1 vediamo sia la Scheda MainMenu che la Finestra Database che mostra l'elenco di tutte le Schede della applicazione.

Nell'elenco delle Macro, invece, spicca, anche perché è casualmente la prima della lista, la AutoExec (fig. 2), che è quella che viene lanciata automaticamente al lancio dell'applicazione e che svolge il compito di Aprire la Scheda MainMenu.

Insomma la Scheda MainMenu, caricata al lancio dell'Applicazione, costituisce il cuore di tutta l'organizzazione dal quale si lanciano tutte le possibili funzionalità, legate ciascuna ad una Macro, e alla quale si torna al termine del lavoro con la singola funzionalità.

La curiosità ci impone di aprire la scheda MainMenu per vedere come viene realizzato il collegamento tra ogni singola Voce di Menu e la corrispondente Macro.

Nella figura 3 vediamo, nell'Editor delle Schede, il Box in cui appare la Lista delle Voci e la sua finestra delle Caratteristiche. In questa possiamo notare come la Lista delle Voci di Menu sia alimentata addirittura da una Tabella (ca-

Figura 1 – Microsoft Access – Una applicazione in dotazione.

Access dispone di un materiale Tutorial molto ricco. Affronteremo la programmazione dapprima vedendo l'applicazione GESTORD, poi realizzando, all'interno della nostra «Applicazione Studio» Stipendi, delle Funzioni, delle Macro e delle Procedure. L'accesso ai vari oggetti di cui è composta l'applicazione Gestord è regolato da una Scheda che fa da Menu e che vediamo sia nel suo aspetto iniziale, sia come voce nell'elenco delle Schede.

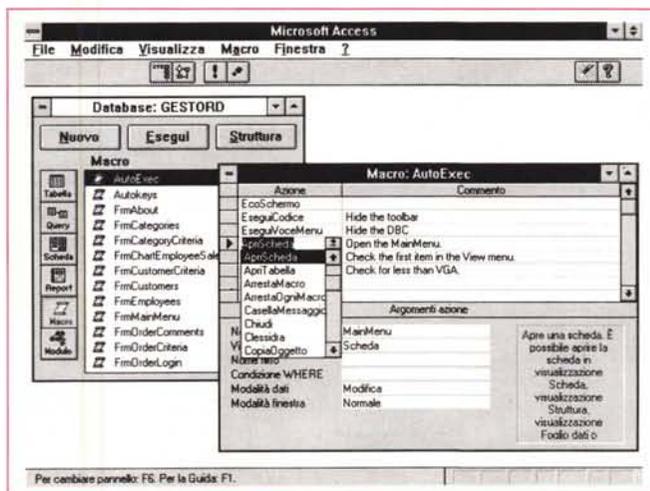
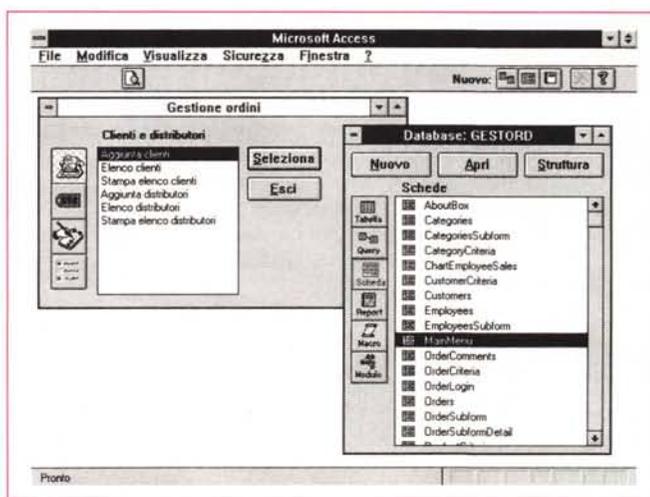
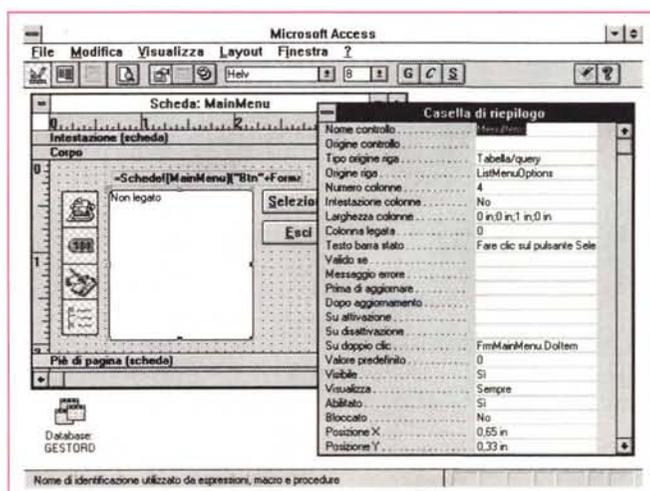


Figura 3 – Microsoft Access – Come è fatta la scheda MainMenu.

La scheda MainMenu ha quattro bottoni. Facendo click su ciascuno dei quattro Bottoni Grafici, nella Box al centro, appare una Lista di voci, collegate ciascuna ad una Macro eseguibile. In pratica si tratta di un Menu con quattro sottomenu, in grado di smistare l'utilizzatore verso le singole Schede, Report o Query. Dalla Finestra della Proprietà relativa alla Box interna al MainMenu constatiamo il fatto che tale Box è collegata ad una Tabella di Menu.

Figura 2 – Microsoft Access – La Macro AutoExec.

Spulciando tra le Macro dell'applicazione Gestord ne appare una caratterizzata dal promettente nome AutoExec. È chiaro che si tratta della Macro che viene eseguita al caricamento dell'applicazione e che dovrebbe servire sia per settare le impostazioni generali dell'applicazione stessa sia per eseguire la prima maschera con il MainMenu, il menu principale dell'applicazione, che è quello visto nella precedente figura.



ratteristica Origine Riga) e come facendo Doppio Click sulla Voce venga eseguita la Procedura Doltem scritta nel modulo FrmMainMenu (FrmMainMenu.Doltem).

Apprendo la Tabella origine dei dati Menus (fig. 4) vediamo come sia semplicissima. Contiene quattro campi che indicano, rispettivamente, menu di primo livello di appartenenza (quello identificato dai quattro bottoni), voce individuale di secondo livello, Testo della Voce di Menu, e infine Macro to Run, ovvero nome della Macro da eseguire in caso di scelta di quella voce di menu.

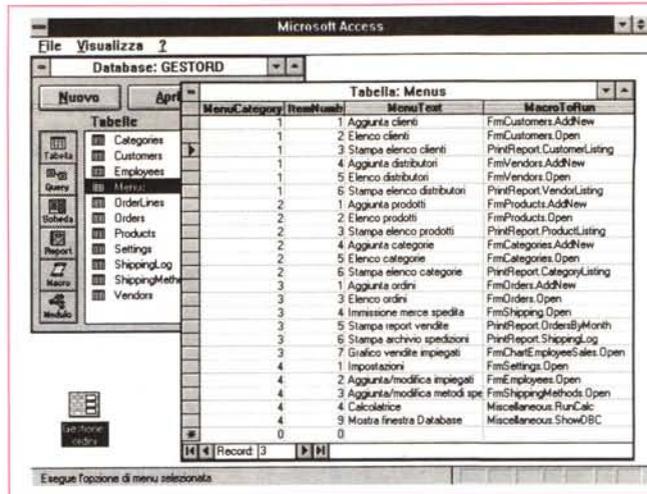


Figura 4 - Microsoft Access - Come è fatta la Tabella Menu. In pratica tutta l'applicazione GESTORD ruota attorno a tale Tabella, facilmente aggiornabile in caso inserimento di una ulteriore sottoprocedura. La MenuCategory corrisponde a uno dei quattro Bottoni, l'ItemNumber è il numero progressivo all'interno della Category. MenuItem è la voce che appare nella Box e nell'ultima colonna troviamo il nome della Macro richiamata da quella voce di menu.

Prima considerazione

Il sistema visto ora rappresenta un modo brillante per assemblare i vari pezzi di un'applicazione Access. Quindi si può ipotizzare uno sviluppo dell'applicazione Access per gradi.

Dapprima si lavora in modalità interattiva, realizzando e testando i vari «pezzi», Tabelle, Query (che svolgono anche compiti tradizionalmente svolti da procedure Batch), Schede e Report.

Poi si realizzano una serie di Macro, anche di «una sola riga», che aprono o lanciano ciascuno degli oggetti.

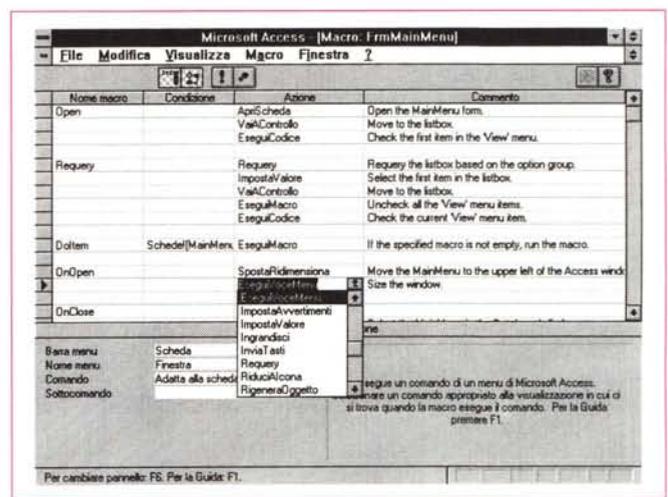
Si può poi creare una Tabella di Menu, in cui si inseriscono tutte le voci e le Macro, e poi una Scheda, che fa da Main Menu, che la gestisca. Il tutto scopiazzando il metodo ora visto.

Se le funzionalità non fossero numerosissime si potrebbe in alternativa ipotizzare una unica Scheda, tutta fatta di bottoni, ognuno dei quali esegue direttamente una Macro.

Qualsiasi sistema si scelga va sottolineato il fatto che la programmazione, in caso di applicazioni di media complessità, può essere affrontata in un secondo momento e soprattutto può essere vista come semplice fatto organizzativo, di oggetti sviluppati e messi a punto sfruttando le modalità interattive.

Figura 5 - Microsoft Access - Ambiente Editor delle Macro.

Ad ogni voce di menu, quindi, deve corrispondere una Macro. Si va dalle Macro più semplici, ad esempio quelle che aprono un oggetto Tabella, Query, Scheda o Report a quelle più complesse che eseguono più passaggi, fino a quelle che lanciano una Procedura, scritta in Access Basic. L'aspetto interessante consiste nella facilità della manutenzione di un sistema del genere, in cui una nuova funzionalità dell'applicazione viene inserita mediante un nuovo record nella Tabella.



Gli ambienti Macro e Modulo

Concludiamo la nostra indagine all'interno dell'applicazione GESTORD vedendo come si presenta l'Editor delle Macro e poi l'Editor dei Moduli.

L'Editor delle Macro (fig. 5) è un ambiente «simil-spreadsheet» nel senso che appare come una Tabella, con in cima la sua brava Barra degli Strumenti, e con alcune colonne, da due a quattro, a seconda che si decida di vedere o meno e di usare o meno la colonna con i Nomi delle Macro e quella con le Condizioni. Le due colonne sempre presenti sono Azione e Commento.

Una Macro è caratterizzata quindi da una o più Azioni, che si scelgono in un elenco di 43 tipi Azioni, e da una serie di caratteristiche proprie della Azione. In figura 6 una Tabella con un piccolo campionario.

L'Editor dei Moduli ha un aspetto più tradizionale, nel senso che occorre scrivere le istruzioni a mano, anche se ci si può avvalere delle funzioni di editing messe a disposizione dalla voce Modifi-

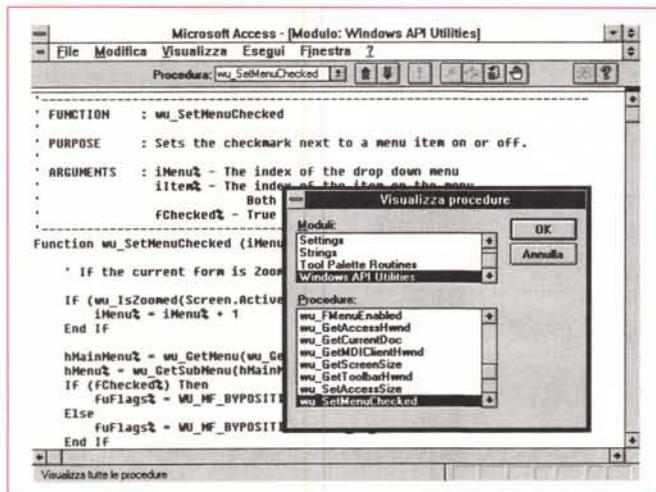
Esempi di Espressioni MACRO ACCESS

| Azione | Caratteristiche | | | | |
|--------------------|--|---|-------------------------|----------------------------|---------------------------|
| ApriScheda | Nome scheda es. Impiegati | Visualizzazione Scheda | Nome filtro | Condizione Where | Modalità dati Modifica |
| Stampa | Intervallo di Stampa es. Tutto | Da pagina | A pagina | Qualità stampa Alta | Copie |
| EseguiCodice | Nome funzione es. Scopi | -- | -- | -- | -- |
| ImpostaValore | Elemento Scheda([MainMenu].Visible) | Espressione | -- | -- | -- |
| EseguiVoceMenu | Barra menu Scheda | Nome menu File | Comando Salva record | Sottocomando | -- |
| EseguiApplicazione | Riga di comando CALC.EXE | -- | -- | -- | -- |
| EcoSchermo | Eco Schermo es. Sì | Testo barra stato Procedura in Corso | -- | -- | -- |
| CaseggiMessaggio | Messaggio es. Attenzione Procedura in Corso | SegnaleAcustico Sì | Tipo Avvertimento ? | Titolo Messaggio Utente | -- |

Figura 6 - Microsoft Access - Esempi di Macro. In questa Tabellina mostriamo alcune delle 43 Azioni inseribili in una Macro. Ognuna di queste dispone di un personale elenco di Caratteristiche che vanno impostate e che indirizzano il dettaglio dell'Azione. Una Macro può essere costituita da una serie di Azioni che vengono eseguite in sequenza.

Figura 7 - Microsoft Access - Come si presenta l'editor delle procedure.

L'ultimo anello della catena è il Modulo che contiene Funzioni o Procedure. Questi due oggetti si concretizzano in un codice che va scritto in un ambiente Editor abbastanza tradizionale caratterizzato da una Toolbar... per programmatori. Vanno organizzate in Moduli, nel senso che un modulo può contenere più Funzioni o Procedure. C'è anche una Window che serve per pescare la procedura desiderata nel suo modulo.



analogia con quanto permesso dal Visual Basic.

Al lavoro con il nostro Caso Studio

Ci proponiamo alcuni semplici e documentabili obiettivi.

Innanzitutto realizzeremo due Funzioni, che useremo in varie situazioni, poi realizzeremo una Procedura più tradizionale, il cui scopo è quello di far scorrere record per record, un archivio. Infine realizzeremo un nostro MainMenu.

Due funzioni

La prima si chiama GRAT, e accetta un argomento logico. Il risultato deve essere 60.000 se GRAT è Vero, 20.000 se Falso.

La sintassi è ben evidente in figura 8 (in alto a destra nella finestra Modulo: Funzioni).

Per scrivere la Funzione occorre attivare il Bottone Modulo nella Finestra Database e nell'Editor che appare occorre cominciare a scrivere Function e il nome della Funzione.

Da notare come sia possibile utilizzare anche istruzioni di tipo logico che, nel nostro caso, servono per pilotare il risultato.

Una volta scritta la funzione va compilata (c'è una specifica voce nel menu Esegui) ed è pronta per l'uso.

Abbiamo usato la nostra Funzione innanzitutto nella definizione della Struttura della Tabella DATIUNO, indicando come valore predefinito del campo GRATIFICA il valore calcolato GRAT(L). In Datiuno c'è infatti un campo Logico L che assume valore Sì/No e che quindi può essere utilizzato per calcolare la Gratifica.

Abbiamo usato la nuova Funzione anche in una Query di tipo Aggiornamento (questa particolarità viene evidenziata da un punto esclamativo accanto al suo nome). Per cui, modificando i valori, quelli che inizialmente erano 20.000 e 60.000 e lanciando la Query, il valore del campo Gratifica viene aggiornato in tutti i record.

Ci riferiamo a questo quando affermiamo che le Query possono servire anche per eseguire le cosiddette elaborazioni Batch, quelle, in parole povere, che eseguono l'aggiornamento di grossi insiemi di dati.

La funzione Grat può poi, lo vedremo dopo, essere utilizzata in una Procedura tradizionale.

La seconda funzione IRPEF è più complessa, soprattutto per il fatto che si appoggia su una Matrice di valori (fig. 9).

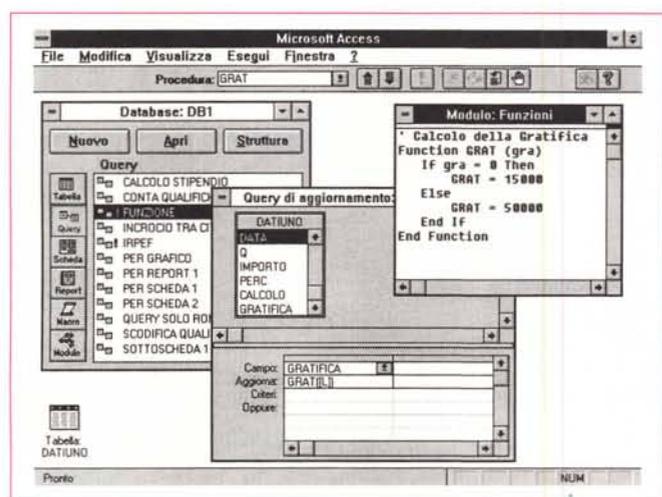
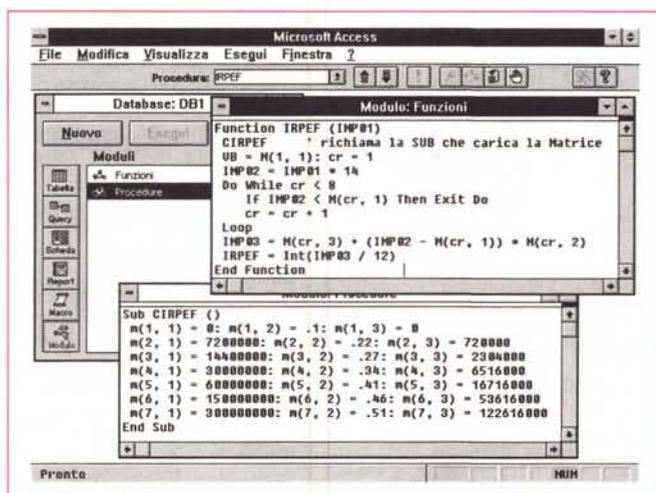


Figura 8 - Microsoft Access - Una funzione usata in una Query.

La forma più semplice di programmazione, ma che è in grado di risolvere un bel po' dei problemi insiti nella applicazione è la possibilità di costruire delle Function aggiuntive rispetto a quelle già disponibili. Tali funzioni possono contenere anche istruzioni logiche, istruzioni di flusso, cicli, ecc. Alla nostra Funzione Gratifica va passato un parametro di tipo logico. Se il parametro è 1 (vero, sì) la Funzione assume il valore 60000, altrimenti assume il valore 20000.

Figura 9 - Microsoft Access - Una Funzione che richiama una Procedura.

Abbiamo realizzato, per il nostro programma di Stipendi, una Funzione che esegue, secondo un algoritmo descritto nell'articolo, il calcolo della ritenuta IRPEF da applicare ad un certo Stipendio. La Funzione si chiama IRPEF e utilizza una Array di dati, in cui sono memorizzate le fasce di Reddito e le Percentuali da applicare per ciascuna fascia (cfr. Istruzioni per il modulo 740).



ca del Menu (fig. 7). I Bottoni sulla Barra servono per passare da una Procedura (un Modulo contiene una o più procedure) all'altra, oppure per imporre la modalità Passo Passo all'esecuzione

della Procedura, ecc. Insomma funzioni di Debug.

Per passare da una Procedura ad un'altra si può usare la comoda finestra Visualizza Procedure. Anche questo in

Serve per calcolare la ritenuta Irpef di un certo Importo. Supponiamo un Importo mensile, che viene moltiplicato per 14 (le mensilità previste nell'anno). Su tale Importo Annuale viene calcolata la ritenuta IRPEF globale, che viene poi divisa per 12 per renderla mensile.

In breve con una subroutine CIRPEF abbiamo caricato la Matrice di valori in base alla quale si calcola la ritenuta Irpef (cfr. Istruzioni del modulo 740 di quest'anno).

La prima istruzione della Funzione IRPEF esegue la subroutine. Va precisato che la Matrice M(x,y) va dichiarata come Globale per poter essere usata in qualsiasi punto dell'applicazione.

Poi con un Ciclo:

DO WHILE <condiz.>

IF <condiz.> THEN EXIT DO

LOOP

viene percorsa la matrice e confrontato il suo primo valore con quello dell'Importo Annuale. Trovato quello che si chiama lo «scaglione di reddito» giusto si esce dal Ciclo, con i valori M(x,y) che servono per il calcolo esatto, che viene puntualmente fatto.

Al pari della funzione GRAT anche la IRPEF può essere usata dovunque.

È evidente che una Funzione, in una applicazione, può sobbarcarsi un gran volume di lavoro (di calcolo) il che rende molto semplice la manutenzione dell'applicazione stessa che può riguardare ad esempio solo la Funzione, o, al contrario, solo il programma che richiama la funzione.

Il nostro Menu ad Icone

sulla falsariga di quanto detto per la procedura GESTORD realizziamo la nostra Scheda Menu. Mettiamo solo quattro Bottoni Grafici ad ognuno dei quali associamo una Macro preconfezionata.

Al di sotto del Menu vediamo la Macro che non fa null'altro che lanciare la Query (vista tre numeri fa) che conta le Qualifiche.

Un menu totalmente grafico può assumere numerose varianti. Ad esempio invece di tanti bottoni grafici è possibile utilizzare un unico disegno nel quale posizionare una serie di «zone» calde, sulle quali fare Click per lanciare le varie sotto applicazioni.

Nelle due figure 10 e 11 vediamo la costruzione del menu e il suo successivo uso. I primi due Bottoni richiamano una Scheda e una Query realizzate negli scorsi numeri.

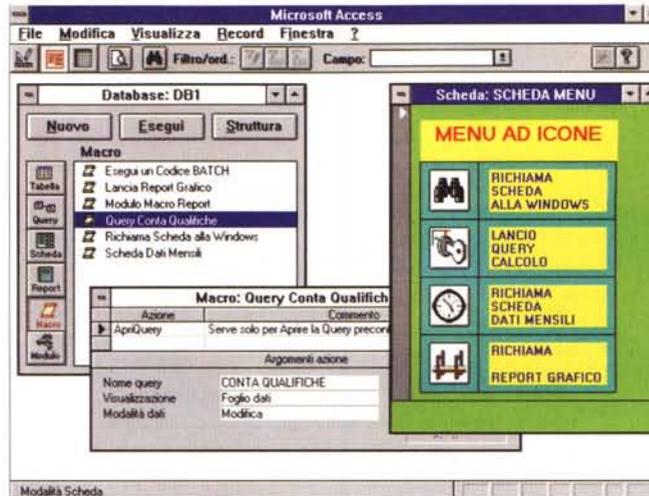


Figura 11 - Microsoft Access - La Macro Macro all'opera.

Il meccanismo, descritto nella didascalia precedente, è semplicissimo. È chiaro che in caso di sviluppo di applicazioni verticali, occorre da una parte rendere i vari oggetti più rigidi (ad esempio non ridimensionabili) e dall'altra modificare l'ambiente, ad esempio togliendo la bottoniera o il menu che non serve all'utilizzatore dell'applicazione.

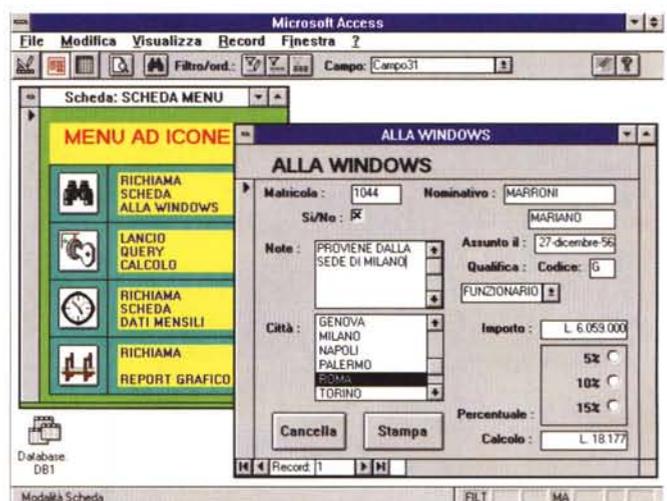


Figura 10 - Microsoft Access - Una Macro-Macro. In questa videata vediamo tra cose. Una Scheda contenente solo dei Bottoni e che quindi usiamo come Menu. La Finestra Database aperta alla pagina Macro, nella quale vediamo i Nomi delle cinque Macro da noi create. E poi una Macro con una sola istruzione di apertura Scheda. In pratica il bottone della Scheda (quello con il binocolo) è associato alla macro Richiama Scheda alla Windows (l'abbiamo costruita nella seconda puntata) che consiste nella sola istruzione ApriScheda.

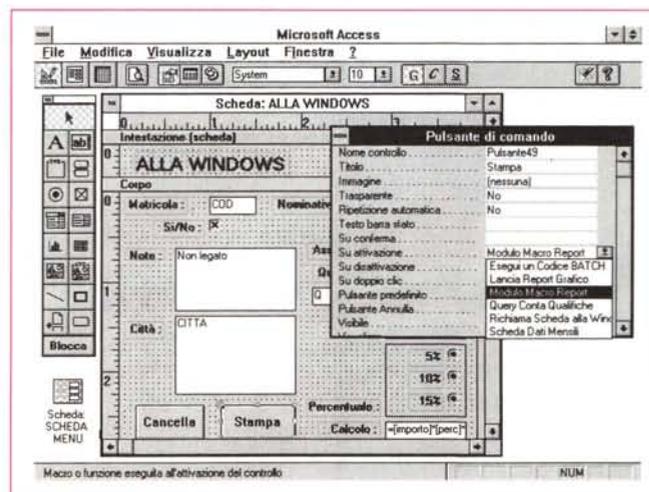


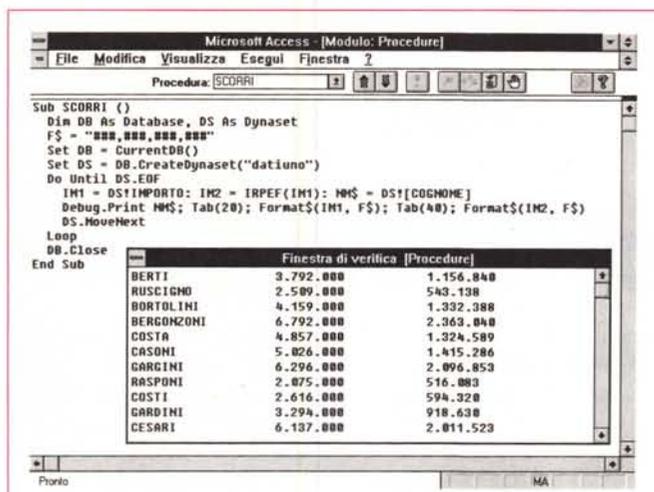
Figura 12 - Microsoft Access - Bottone che lancia una Macro. Programmazione a Oggetti. Un Bottone, prelevato dal Pannello degli Strumenti, posizionato su una Scheda che richiama una Macro. L'associazione della Macro al Bottone è assicurata dalla caratteristica «Su attivazione» del Bottone.

Un Bottone e una Macro

Sono fatti uno per l'altra. Ebbene abbiamo visto come sia pos-

sibile costruire delle Schede piene di Bottoni che assumono il compito di fare da Menu. È anche possibile inserire in una qualsiasi Scheda dei Bottoni per facilitare il lancio di qualche funzionalità

Figura 13 – Microsoft Access – Una procedura Batch più tradizionale. Questo è il listato che più si avvicina al modo tradizionale di programmare. Si tratta di un programma di scorrimento di una Tabella, record per record. Per ogni record vengono visualizzati, nella comoda finestra di Debug (usata solo per controllare il buon funzionamento del programma), alcuni dati. L'aspetto più interessante dell'Access Basic è costituito dalla possibilità di definire delle speciali variabili «globali», indicanti l'intero Database, l'intera Tabella, ecc.



Ma si possono visualizzare anche dati dell'archivio. Es.

? DS! COGNOME
ROSSI

Oppure ci si può muovere nell'archivio attivo. Es.

DS.MOVENEXT
? DS! COGNOME
BIANCHI

Inoltre poiché si possono inserire più istruzioni in una riga (separate dal carattere «RF») si possono realizzare dei Cicli. Es.

FOR I=1 TO 10: ? DS! COGNOME :DS.MOVENEXT:NEXT I

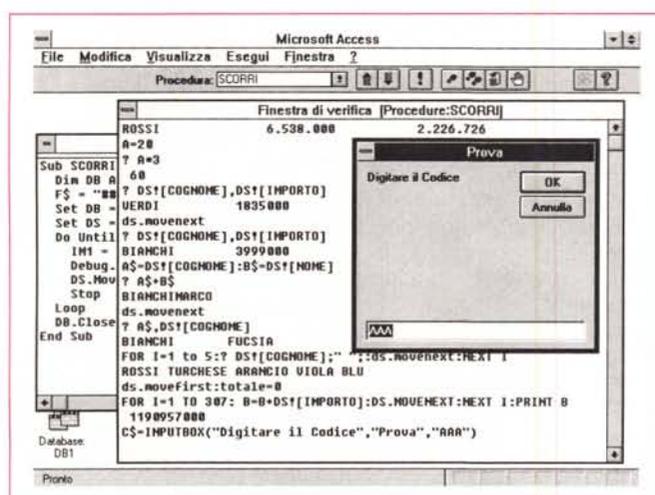


Figura 14 – Microsoft Access – Uso intensivo della Finestra di Verifica.

Nella Procedura Scorri abbiamo inserito una Istruzione Stop, il cui scopo è, ovviamente, quello di bloccare l'esecuzione e di permetterci un «giretto» nella Finestra di Verifica, nella quale è possibile digitare a mano delle Istruzioni che vengono immediatamente eseguite. Purtroppo non tutte le istruzioni di Access Basic sono consentite nella Finestra di Verifica.

Insomma prima di avventurarsi nella scrittura di una Procedura ne possiamo facilmente testare, nell'ambiente interattivo costituito dalla Finestra di Verifica, le Istruzioni.

Conclusioni

Anche vedendo questi ultimi esercizi non si può non rimanere affascinati dalla ricchezza di possibilità offerte dall'Access.

Potentissimo nella gestione interattiva dei Database. Sostanzioso ambiente di programmazione Object Oriented. Linguaggio Basic orientato alla gestione dei Database. Prodotto rispettoso di tutte le regole vigenti in Windows.

Il pericolo è che in questa ricchezza di possibilità (per fare la stessa cosa ci sono una dozzina di varianti) ci si possa perdere.

L'altro pericolo, connesso a quello appena citato, è quello che l'utilizzatore si lasci influenzare dalle sue conoscenze pregresse. E poiché l'utilizzatore può essere un utente evoluto (provenienza Spreadsheet) oppure un programmatore (provenienza Basic) oppure un esperto in DBMS (provenienza xBase) le possibili strade percorse per raggiungere lo stesso obiettivo saranno effettivamente le più disparate.

È un argomento che andrebbe approfondito.

Tenendo inoltre conto del fatto che non c'è più spazio per parlare di Access nei confronti sia del dBase (abbiamo tradito anche il titolo dell'articolo), sia degli altri prodotti nel frattempo apparsi sulla scena, vi promettiamo un successivo articolo che tratti questi due aspetti.

accessoria. L'oggetto, al solito, si preleva dal Pannello degli Strumenti e si posiziona nella Scheda. L'aggancio alla Macro è una semplice caratteristica (su attivazione) del Bottone (fig. 12).

Una procedura tradizionale

La vediamo in figura 13. Si tratta di una procedura di scorrimento, nel senso che attiva una Tabella e ne scorre i Record ad uno ad uno. Si chiama SCORRI.

La struttura DO .. LOOP è classica, ma contiene anche delle comode istruzioni con le quali vengono definite variabili legate ad elementi del Database.

```
SET DS = DB.CREATEDYNASET("datiuno")
DO UNTIL DS.EOF
..
DS.MOVENEXT
LOOP
DS.CLOSE
```

Esistono una serie di istruzioni che si legano all'oggetto, Tabella o Query che

sia, definito come oggetto attivo e che lo rendono in tal modo facilmente maneggiabile.

Una comoda finestra. La finestra di Verifica

Il Basic, del quale Access Basic è una ulteriore variante, è un linguaggio tradizionalmente interprete. È quindi possibile scrivere direttamente delle istruzioni e vederle immediatamente eseguite.

L'ambiente di Access in cui può praticare questa attività è la Finestra di Verifica, richiamabile ad esempio quando si stia provando una Procedura.

Per provare qualche istruzione al volo abbiamo bloccato, con una istruzione Stop, il programma Scorri, abbiamo aperto la finestra di Verifica e abbiamo provato qualche comando (fig. 14).

L'istruzione «RF» è quella che visualizza il risultato nella Finestra di Verifica. Ad es.

? *3*
6