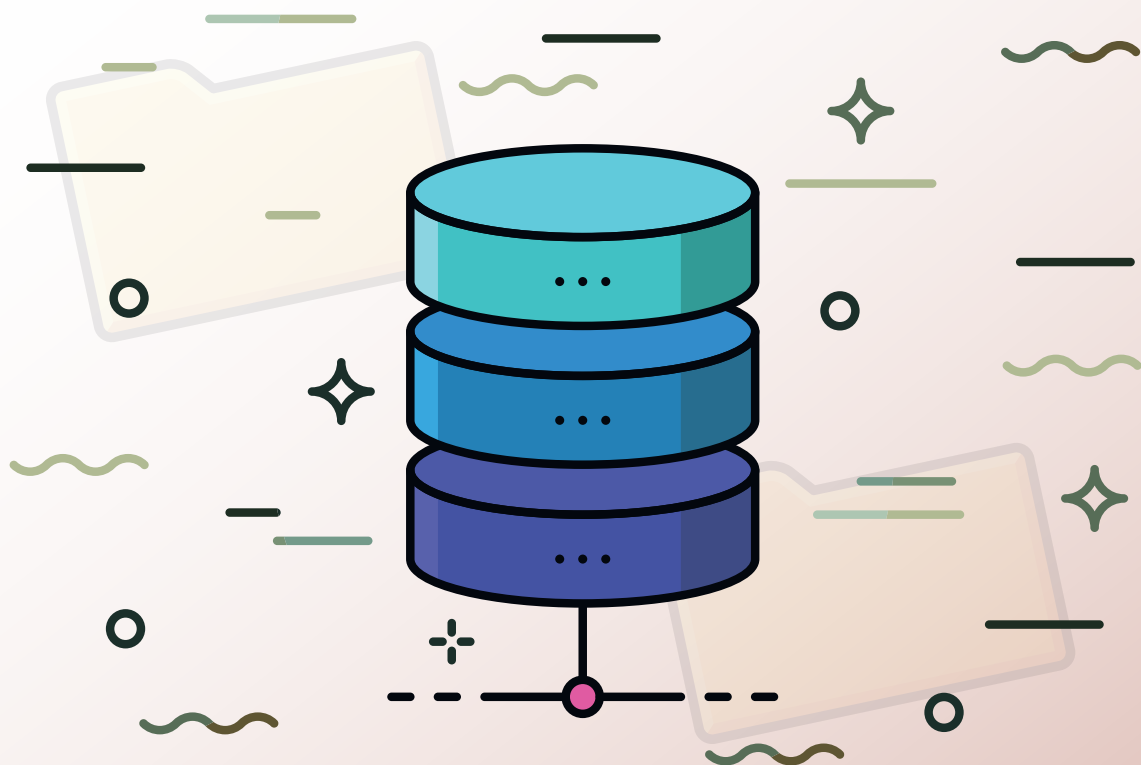


- Alessandra Salvaggio -

Microsoft Access 2019

Programmazione VBA



Nozioni e suggerimenti per la programmazione VBA >>

Creazione di componenti aggiuntivi >>

Sintassi, ottimizzazione, controllo del codice >>

Progetti pratici illustrati passo passo >>

***pro**
DigitalLifeStyle

*pro
DigitalLifeStyle

Microsoft

Access 2019

Programmazione VBA

Alessandra Salvaggio

EDIZIONI
LSWR

Microsoft Access 2019 | Programmazione VBA

Autore: Alessandra Salvaggio

Collana: ^{*pro} DigitalLifeStyle

Publisher: Marco Aleotti

Progetto grafico: Roberta Venturieri

Immagine di copertina: © Andrew Krasovitchii | Shutterstock

© 2019 Edizioni Lswr* - Tutti i diritti riservati

ISBN: 978-88-6895-797-1

eISBN: 978-88-6895-798-8

I diritti di traduzione, di memorizzazione elettronica, di riproduzione e adattamento totale o parziale con qualsiasi mezzo (compresi i microfilm e le copie fotostatiche), sono riservati per tutti i Paesi. Le fotocopie per uso personale del lettore possono essere effettuate nei limiti del 15% di ciascun volume dietro pagamento alla SIAE del compenso previsto dall'art. 68, commi 4 e 5, della legge 22 aprile 1941 n. 633.

Le fotocopie effettuate per finalità di carattere professionale, economico o commerciale o comunque per uso diverso da quello personale possono essere effettuate a seguito di specifica autorizzazione rilasciata da CLEARedi, Centro Licenze e Autorizzazioni per le Riproduzioni Editoriali, Corso di Porta Romana 108, 20122 Milano, e-mail autorizzazioni@clearedi.org e sito web www.clearedi.org.

La presente pubblicazione contiene le opinioni dell'autore e ha lo scopo di fornire informazioni precise e accurate. L'elaborazione dei testi, anche se curata con scrupolosa attenzione, non può comportare specifiche responsabilità in capo all'autore e/o all'editore per eventuali errori o inesattezze.

L'Editore ha compiuto ogni sforzo per ottenere e citare le fonti esatte delle illustrazioni. Qualora in qualche caso non fosse riuscito a reperire gli aventi diritto è a disposizione per rimediare a eventuali involontarie omissioni o errori nei riferimenti citati.

Tutti i marchi registrati citati appartengono ai legittimi proprietari.

EDIZIONI
LSWR

Via G. Spadolini, 7
20141 Milano (MI)
Tel. 02 881841
www.edizionilswr.it

Printed in Italy

Finito di stampare nel mese di ottobre 2019 presso "Rotomail Italia" S.p.A., Vignate (MI)

(*) Edizioni Lswr è un marchio di La Tribuna Srl. La Tribuna Srl fa parte di LSWR GROUP.

Sommario

INTRODUZIONE A VBA	9
Ambiente di lavoro.....	10
Gli elementi di un progetto Visual Basic	11
Convenzioni per i nomi	13
Modello a oggetti	13
Livello di protezione.....	15
Il database ScuolaStella.....	17
Un'applicazione database	21
Autore e utente.....	21
1. CREARE UN NAVIGATORE PERSONALIZZATO	23
Migliorare l'aspetto dei pulsanti.....	31
2. DISABILITARE ALCUNI PULSANTI.....	33
I commenti	37
Scrivere il codice con ordine.....	38
3. RISOLVERE UN PICCOLO PROBLEMA: GLI OPERATORI AND E OR	41
4. COPIARE I CONTROLLI E IL CODICE DA UNA MASCHERA ALL'ALTRA.....	43
5. ASSEGNARE IL FUOCO AL CONTROLLO CORRETTO	45
6. UNA MASCHERA IN SOLA LETTURA.....	49
7. TERMINARE LE MODIFICHE ALLA MASCHERA	55
8. CREARE UN MESSAGGIO PERSONALIZZATO IN RISPOSTA A UN'AZIONE DELL'UTENTE	59

9.	FILTRARE I DATI DI UNA MASCHERA	65
10.	MOSTRARE VISIVAMENTE CHE I RECORD SONO FILTRATI.....	73
11.	FILTRARE LA MASCHERA IN MODALITÀ DI SOLA LETTURA	81
12.	CREARE UNA MASCHERA A ORIGINE MULTIPLA	85
	Casella combinata di ricerca	93
	Sincronizzare la casella combinata.....	97
	Distinguere visivamente la casella combinata	97
13.	ATTIVARE E DISATTIVARE LE SOTTOMASCHERE	99
	Casella di ricerca	101
14.	CONTARE I RECORD DI UNA SOTTOMASCHERA	103
15.	INVIARE UN'E-MAIL	105
16.	LE VARIABILI.....	111
17.	INTERAGIRE CON L'UTENTE	117
18.	SALVARE UN OGGETTO DEL DATABASE IN PDF.....	121
19.	GESTIRE GLI ERRORI	123
20.	CREARE VARIABILI GLOBALI.....	127
21.	VERIFICARE CHE L'UTENTE ABBA COMPILO TUTTA LA MASCHERA	133
	Costanti che rappresentano i controlli disponibili	139
22.	MOSTRARE UN DATO CON UNA BARRA	141
	Aprire una maschera con un filtro	145
23.	UNA FINESTRA DI ZOOM.....	149
24.	ATTIVARE E DISATTIVARE UN CONTROLLO	153
	Mostrare un messaggio in base al contenuto di una casella di testo	156
25.	ADO	159
	Scrivere nella tabella CAP	163
26.	VERIFICA DELLA PARTITA IVA	167

27. VERIFICA DEL CODICE FISCALE.....	173
28. VERIFICA DEL CODICE IBAN.....	181
29. I REPORT	187
30. CAMBIARE COLORE AGLI OGGETTI GRAFICI	193
31. INTERAZIONE CON MICROSOFT WORD	199
Recuperare il percorso del file Access	213
32. UN GRUPPO DI OPZIONI.....	215
33. APPLICAZIONE DATABASE	223
Avviare una maschera all'apertura del database	228
Nascondere la barra multifunzione	230
La maschera di apertura centrata	230
Sostituzione dello schermo di apertura.....	232
34. CREARE UN'APPLICAZIONE DATABASE CON MASCHERE DI SPOSTAMENTO	233
Correggere i pulsanti per l'invio di e-mail e il salvataggio di report	236
Un pulsante per stampare i report.....	238
Rendere visibili i colori dei rettangoli sul report rptMedieFinali.....	241
35. ANCORA SULL'INVIO DI E-MAIL.....	245
Verificare la correttezza dell'indirizzo e-mail.....	251
36. IL DEBUG DEL CODICE.....	253
Gli strumenti di debugging	255
37. MIGLIORARE L'EDITOR	259
I blocchi di commento.....	259
Smart Indenter	261
38. DAO.....	265
Il modello a oggetti DAO.....	266
Lavorare con le tabelle con TableDef	269
Lavorare con le relazioni.....	284
Lavorare con le query	290
Recordset.....	292
Aprire un database esterno	299

39. ADO - ADOX	303
Il modello a oggetti ADO	304
Il modello a oggetti di ADOX	305
Lavorare con le tabelle	306
Lavorare con le relazioni	315
Lavorare con le query	317
Recordset.....	319
Aprire un database esterno	323
INDICE ANALITICO	327

Introduzione a VBA

La suite da ufficio **Microsoft Office** è sicuramente uno dei software più diffusi al mondo; molti ne apprezzano le **potenzialità**, ma tanti ignorano che è possibile **estenderne** le capacità per adeguarle alle proprie necessità. Tutto questo è possibile grazie **VBA** o **Visual Basic for Applications**, un linguaggio di programmazione appartenente alla famiglia del **Visual Basic**.

Chi conosce Visual Basic si troverà a proprio agio con Visual Basic for Applications, in tutto affine al fratello maggiore; l'unica differenza è data dalle istruzioni specifiche per operare sui prodotti Office.

Anche chi è alla prima esperienza di programmazione potrà trovare in questo libro degli spunti interessanti per incrementare le capacità e la flessibilità dei suoi database. Si chiede solamente una discreta conoscenza di Access e dei database relazionali, per il resto le istruzioni saranno date passo passo, in modo da guidare tutti a realizzare i progetti proposti dal libro. In particolare, ci occuperemo di scrivere codice VBA per migliorare le funzionalità delle maschere e dei report.

Sul sito Internet connesso al libro (<http://www.sos-office.it/libro-access2019vba.html>) troverete il database ScuolaStella.accdb che contiene le tabelle e i dati su cui lavorare e il file ScuolaStella_svolto.accdb con tutti gli esercizi proposti dal libro svolti.

Questo testo si basa sulla versione 2019 di Office, ma la maggior parte degli esempi proposti funziona correttamente anche con le versioni precedenti. Dove ci sono differenze significative fra le diverse versioni lo segnaleremo.

Ambiente di lavoro

Naturalmente avrete a che fare con il database Access, con il quale dovrete già avere familiarità, ma ancora di più con l'editor di Visual Basic, a cui dedicheremo le prossime pagine. Ricordate che tutto il codice VBA che scriverete vive unicamente all'interno del programma Office, nel nostro caso Access, che state utilizzando. L'unico file con cui avrete a che fare sarà il file .accdb che contiene sia i dati, le tabelle e le query, sia gli oggetti come le maschere, i report e le pagine, sia il codice VBA associato a questi oggetti.

Per aprire l'editor di Visual Basic, all'interno di Access (del resto non può vivere all'esterno delle applicazioni Office), innanzitutto create un nuovo database, poi portatevi alla scheda **Strumenti database** della barra multifunzione. Nel gruppo **Macro**, premete il pulsante **Visual Basic**.

Se, invece, lavorate con le vecchie versioni di Access, scegliete **Strumenti > Macro > Visual Basic Editor**, oppure premete contemporaneamente ALT + F11.

L'editor di Visual Basic (Figura 0.1) si compone di tre sezioni principali:

1. **Area del codice:** la prima volta che avviate l'editor, quest'area si presenta vuota. In seguito, qui vedrete e scriverete il codice VBA.
2. **Area del progetto:** in questa finestra vedete tutti gli elementi che costituiscono un progetto. A ciascuno degli oggetti presenti nel progetto può essere associato del codice (ne parleremo nel paragrafo successivo).
3. **Area delle proprietà** (qualora non vedeste questa finestra, scegliete **Visualizza > Finestra Proprietà**): qui trovate le proprietà di tutti gli elementi che avete inserito nel vostro progetto. Discuteremo delle proprietà dei singoli elementi più avanti, man mano che li utilizzeremo. Qui accenneremo solo alla proprietà più importante, ossia **Name**. Attraverso questa proprietà ogni oggetto viene identificato in maniera univoca. È molto importante che assegniate dei nomi significativi a tutti gli oggetti del vostro progetto, solo così potrete richiamarli in modo semplice e, soprattutto nei progetti complessi, ricordarvi la funzione di ciascun elemento. Più avanti discuteremo delle convenzioni da adottare per assegnare i nomi agli oggetti.

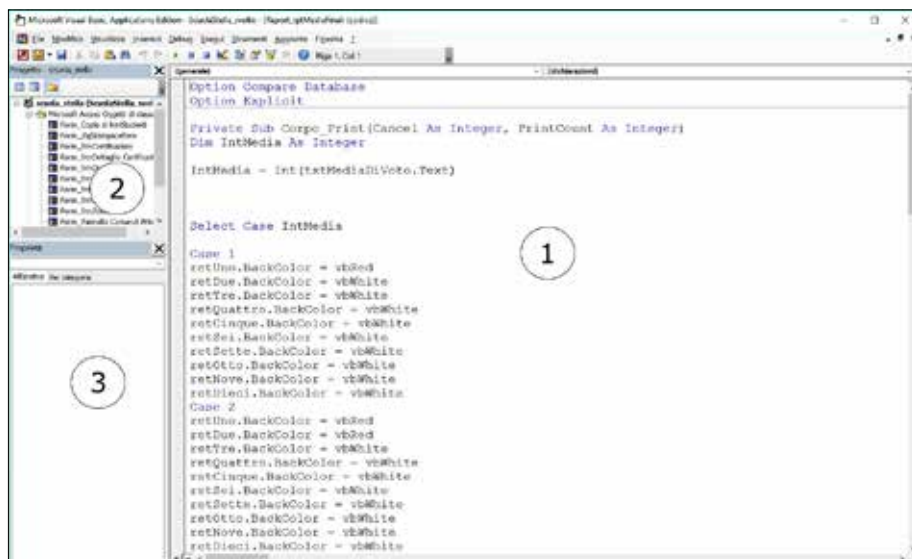


Figura 0.1 - L'editor di Visual Basic.

Gli elementi di un progetto Visual Basic

Un progetto VBA si compone di numerosi elementi a ciascuno dei quali può essere associato del codice Visual Basic. La Figura 0.2 mostra l'**Area del progetto** relativa a un progetto composto da vari elementi. Li analizzeremo uno per uno.

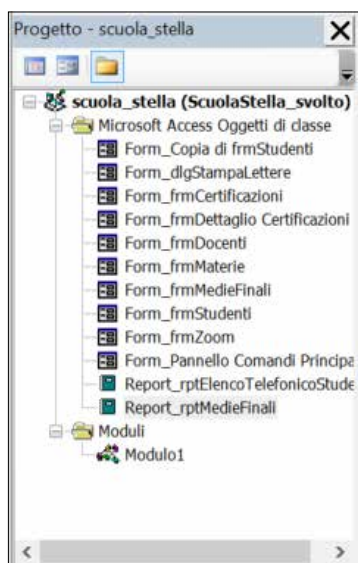


Figura 0.2 - La finestra del progetto.

Innanzitutto, troviamo il file di database aperto, in questo caso ScuolaStella_svolto, che contiene tutti gli altri elementi. Tra gli elementi contenuti nel database troviamo le sue maschere (Form_frmStudenti, Form_MedieFinali ecc.) e i suoi report (rptElencoTelefonicoStudenti ecc.). Questi oggetti, nella finestra del progetto, sono contraddistinti da un nome costruito con un prefisso che indica il tipo di oggetto (form, che è il nome inglese delle maschere, e report), un trattino basso o underscore ("_") e il nome vero e proprio assegnato all'oggetto all'interno di Access.

A ogni maschera e report è associato un foglio di codice, che normalmente ospita le istruzioni VBA necessarie al funzionamento della maschera o del report stesso.

Sia sulle maschere che sui report sono presenti i controlli che permettono di visualizzare e formattare i dati o eseguire delle operazioni. Sicuramente conoscete i controlli che possono essere inseriti in una maschera o in un report di Access; vediamo i più usati:

- **Caselle di testo** o **TextBox**: qui l'utente può digitare del testo;
- **Etichette** o **Label**: spazi in cui potete scrivere del testo. Per esempio, per spiegare all'utente cosa deve scrivere in una casella di testo;
- **Caselle di controllo** o **CheckBox** : caselle rettangolari, di solito presentate in gruppi, in cui l'utente può selezionare **più di una** opzione;
- **Pulsanti di opzione** o **Option Button** : pulsanti circolari, di solito presentati in gruppi, in cui l'utente può selezionare **una sola** opzione;
- **Caselle combinate** o **Combo Box**: caselle di testo unite a elenchi a discesa in cui l'utente può selezionare l'opzione che desidera o digitare del testo. Nelle applicazioni Office, un esempio di casella combinata è quella utilizzata per la scelta del tipo di font con cui si vuole scrivere;
- **Liste** o **ListBox**: una casella di riepilogo con un elenco di dati fra cui l'utente può scegliere;
- **Pulsanti di comando**: normali pulsanti, premendo i quali si avvia una azione.

A ognuno di questi controlli può essere associato del codice.

Oltre agli oggetti di cui abbiamo parlato fin ora, un progetto VBA può contenere anche dei **Moduli**, raggruppati nella cartella Moduli.

Un modulo non può contenere controlli come le maschere e i report, ma contiene esclusivamente codice. In genere si tratta di porzioni di codice generali che possono essere richiamate e utilizzate in più punti del progetto (se il concetto non vi è chiaro, non preoccupatevi troppo, prima della fine del libro capirete...).

Convenzioni per i nomi

Prima di cominciare a lavorare con VBA, occorre precisare che, anche se all'interno di una applicazione Office, si tratta sempre di programmazione e l'ordine e la precisione sono molto importanti se non si vogliono fare pasticci.

La prima regola fondamentale è quella di assegnare a ciascun elemento del vostro progetto un nome "parlante", ossia un nome che, anche a distanza di tempo, ci permetta di capire con che tipo di oggetto abbiamo a che fare e a che cosa serve. Credete, non è una cosa da poco quando si ha a che fare con molti oggetti e controlli.

Normalmente si adotta un sistema di denominazione composto da un prefisso che indichi il tipo di oggetto e un nome vero e proprio che descriva l'oggetto. Fra il prefisso e il nome non può esserci lo spazio vuoto (non è ammesso da VBA, così come non sono ammessi i segni di punteggiatura): per questo, per identificare velocemente le due parti, si scrive la prima lettera del nome vero e proprio con la lettera maiuscola.

Facciamo un semplice esempio: una casella di testo in cui un utente deve inserire il suo cognome si dovrebbe chiamare `txtCognome`, una casella combinata in cui si sceglie la provincia di residenza si chiamerà `cboProvincia`, una maschera si chiamerà `frm` seguito dal nome della tabella in cui legge e scrive. Semplice no? È bene abituarsi fin dall'inizio a nominare correttamente gli oggetti, tutto poi vi verrà più facile. Qui di seguito troverete un elenco dei principali prefissi in uso (tenete presente che i prefissi sono tratti dai nomi inglesi dei diversi controlli e oggetti):

- **frm**: per le maschere
- **rpt**: per i report
- **mod**: per i moduli
- **txt**: per le caselle di testo
- **lbl**: per le etichette
- **cbo**: per le caselle combinate
- **lst**: per le liste
- **cmd**: per i pulsanti di comando
- **opt**: per i pulsanti di opzione
- **chk**: per le caselle di controllo

Modello a oggetti

Occorre fare una piccola premessa sui principi su cui si fonda VBA, per poter fissare quella terminologia che poi utilizzeremo nel corso del libro. VBA è un **linguaggio a eventi** che segue (almeno in parte) i principi della programmazione **orientata agli oggetti**. Cerchiamo di chiarire uno per volta questi due concetti. Un linguaggio di programmazione a eventi è un linguaggio in cui le azioni vengono compiute quando si verifica un

evento. In parole povere, il codice viene eseguito quando l'utente fa qualcosa, come premere un pulsante, scrivere in una casella di testo, uscire da una casella di testo... Tutte queste azioni sono eventi. A ogni evento può essere associato un codice, come dire, all'evento "bussare" viene attivata l'azione "aprire".

Ogni evento è associato a un oggetto (e qui passiamo al secondo concetto): questo significa che non diremo "io busso alla porta" ma "porta_bussano". Qui abbiamo anticipato un po' la sintassi di VBA: infatti, il nome di un oggetto e l'evento sono separati da un trattino basso o, in inglese, underscore ("_").

Ma quali sono gli oggetti? Gli oggetti sono le maschere, i report, i controlli come caselle di testo, pulsanti ecc.

Gli oggetti, oltre a essere associati a degli eventi, possono avere delle proprietà, ossia delle caratteristiche, che possiamo leggere e modificare. Per esempio, l'oggetto pulsante di comando ha come proprietà `caption` che rappresenta il testo che viene visualizzato sul pulsante. In VBA dirò che `cmdChiudi.caption = "Chiudi maschera"`, che vuol dire che la proprietà titolo dell'oggetto pulsante `cmdChiudi` è "Chiudi maschera", che è un po' come se dicessi `capelli.colore = "biondi"`, ossia i miei capelli sono biondi. Ogni proprietà è unita all'oggetto a cui si riferisce da un punto.

Oltre alle proprietà, gli oggetti hanno dei metodi, ossia delle azioni che vengono compiute sugli oggetti stessi. Per esempio, l'oggetto casella di testo dispone del metodo `SetFocus`, che fa in modo che la casella di testo riceva il fuoco, ossia il cursore sia presente nella casella di testo e il testo digitato dall'utente sia inserito nella casella stessa: dirò `txtCognome.SetFocus`, per indicare a VBA di assegnare il fuoco alla casella di testo `txtCognome` (in seguito al verificarsi di un evento che avremo specificato), che è un po' come dire `capelli.taglia`. Come vedete, anche i metodi sono uniti agli oggetti da un punto.

Questa situazione può essere complicata dal fatto che gli oggetti possono essere contenuti gli uni dentro gli altri: `frmAnagrafica.txtCognome.text = "Salvaggio"` significa che la proprietà `text` (ossia contenuto testuale) della casella di testo `txtCognome`, all'interno della maschera `frmAnagrafica`, è uguale a Salvaggio, un po' come dire `Alessandra.capelli = "biondi"`, cioè i capelli della persona Alessandra sono biondi.

La gerarchia degli oggetti viene descritta da sinistra a destra, ovvero l'oggetto citato per primo, quello più a sinistra (`frmAnagrafica` nell'esempio precedente) è quello che contiene gli altri oggetti (`txt.Cognome`), quelli più a destra.

Non sempre bisogna elencare tutti gli oggetti uno dentro l'altro: per esempio, se ci si trova all'interno del modulo associato alla maschera `frmAnagrafica` non ci sarà bisogno di scrivere `frmAnagrafica.txtCognome.text = "Salvaggio"`, ma basterà `txtCognome.text = "Salvaggio"`.

Per concludere il nostro discorso sugli oggetti, precisiamo che all'interno di un progetto Access troverete oggetti con il nome al singolare (per esempio, form) e altri con il nome al plurale (per esempio, forms). Mentre i primi rappresentano un singolo oggetto (per esempio una specifica maschera), i secondi non sono veri e propri oggetti, ma **insiemi** che raccolgono tutti gli oggetti di un determinato tipo. Per esempio, `Forms` è l'insieme che raccoglie tutti gli oggetti `form` del progetto, cioè tutte le sue maschere.

Livello di protezione

Per poter utilizzare il codice VBA all'interno di un file Access, occorre impostare correttamente il livello di protezione desiderato. L'esecuzione di codice che, di fatto, compie delle operazioni all'insaputa dell'utente, può essere pericolosa e non di rado si sono trovati documenti Office che, all'interno del VBA, nascondevano dei virus.

Per questo è possibile fare in modo che Access blocchi l'esecuzione di tutto il codice ma, avendo bisogno del VBA, si tratta di un'impostazione troppo drastica. D'altro canto, lasciare che Access esegua qualsiasi codice, anche quello non scritto da voi, può essere pericoloso. La soluzione giusta consiste nel fare in modo che, all'apertura di tutti i file che contengono codice, Access vi chieda se eseguirlo o meno.

Per impostazione predefinita, Access fa proprio questo e, quando aprite un file, vi avvisa che il contenuto è potenzialmente pericoloso e vi chiede se volete abilitare quel contenuto (Figura 0.3).



Figura 0.3 - Access avvisa che il database include contenuto potenzialmente pericoloso.

Nel caso di codice scritto da voi, quindi sicuro, scegliete tranquillamente **Abilita il contenuto**.

Se volete cambiare le impostazioni di base, aumentare o diminuire (questo ve lo sconsiglio) il livello di protezione, bisogna lavorare nella finestra **Opzioni (File > Opzioni)**. Portatevi alla sezione **Centro di protezione** e premete il pulsante **Impostazioni centro di protezione**. Access vi mostrerà la finestra **Impostazioni centro di protezione**: qui portatevi alla sezione **Impostazioni delle macro** (Figura 0.4).

NOTA

In questo caso, e solo in questo caso, quando Access parla di macro intende istruzioni VBA.

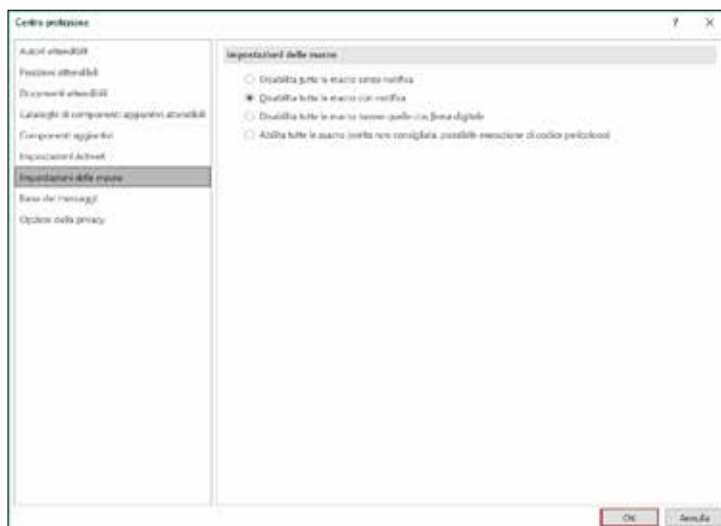


Figura 0.4 - Modificare il livello di protezione per il codice VBA.

Le opzioni disponibili sono:

- **Disattiva tutte le macro senza notifica:** utilizzare questa impostazione se le macro non sono considerate attendibili. Tutte le macro nei documenti e gli avvisi di protezione per le macro verranno disattivati;
- **Disattiva tutte le macro con notifica:** si tratta dell'impostazione predefinita. Utilizzarla se si vogliono disattivare le macro ma si desidera anche ricevere messaggi di avviso nel caso delle macro siano presenti. In questo modo è possibile scegliere se attivare o meno le singole macro;
- **Disattiva tutte le macro tranne quelle con firma digitale:** questa impostazione è uguale all'opzione precedente con la differenza che la macro può essere eseguita se è firmata da un editore attendibile che è già stato considerato tale. In caso contrario, si riceverà una notifica e sarà possibile scegliere se attivare le singole macro firmate o considerare attendibile l'editore. Tutte le macro senza firma vengono disattivate senza notifica;
- **Attiva tutte le macro:** tutte le macro vengono eseguite senza che ne siate avvisati. È un'impostazione pericolosa e vi consigliamo di adottarla.

Noi vi consigliamo di lasciare l'opzione predefinita, **Disattiva tutte le macro con notifica**.

Access dispone anche di una funzione che, mentre svolgete gli esercizi del libro, potrà rivelarsi molto utile per evitare di dover attivare ogni volta il codice VBA.

Infatti, potete decidere di salvare tutti i file con il codice VBA in una cartella e definire quella cartella come **percorso attendibile**.

Nella finestra della **Impostazioni centro di protezione** scegliete la categoria **Posizioni attendibili** (Figura 0.5).

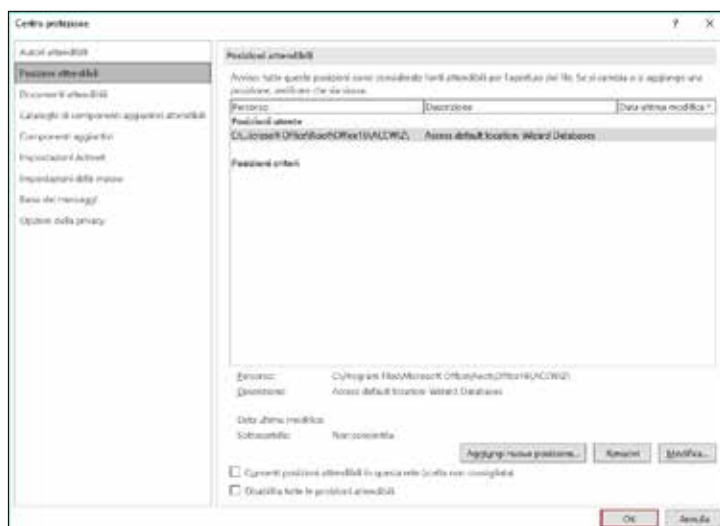


Figura 0.5 - La finestra Protezione alla scheda Posizioni attendibili.

Premete il pulsante **Aggiungi nuova posizione** e individuate la cartella che contiene i file con il codice VBA.

Il percorso comparirà nella parte alta della scheda **Posizioni attendibili** (Figura 0.5). Chiudete la finestra.

Ora potrete tranquillamente aprire i file di questa cartella e il codice VBA sarà automaticamente attivato.

Il database ScuolaStella

Per gli esempi del libro ci serviremo di un database per la gestione dei corsi e degli studenti di una scuola. Si tratta di un database composto di sette tabelle di cui vedete lo schema delle relazioni nella Figura 0.6.

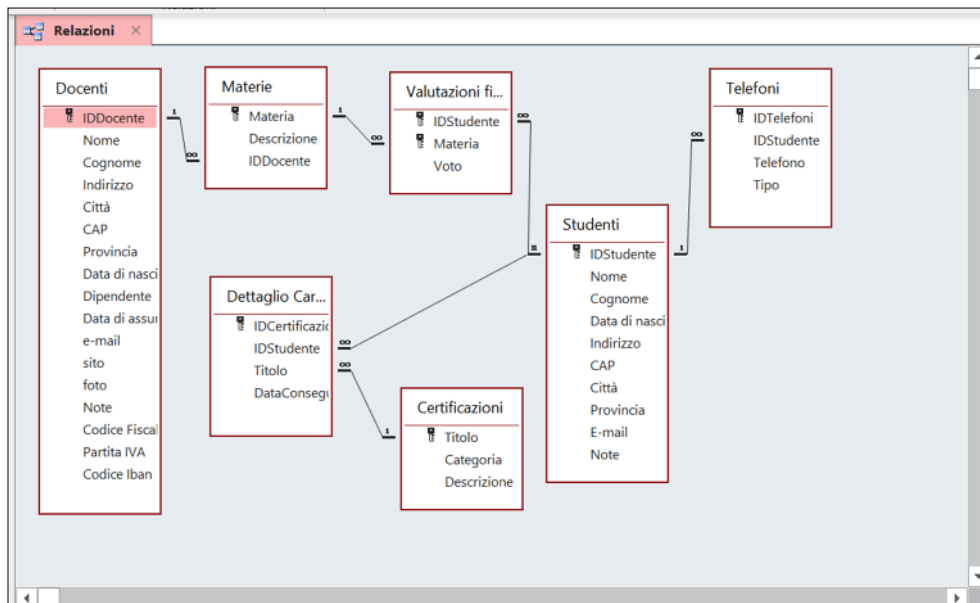


Figura 0.6 - Le relazioni fra le tabelle del database Scuola_stella.

Di seguito descriveremo le tabelle che compongono il database, incominciando dalla tabella *Studenti*.

Studenti

Nome campo	Tipo dati	Nota
IDStudiante	Numerazione automatica	Chiave primaria
Nome	Testo	
Cognome	Testo	
Indirizzo	Testo	
CAP	Testo	
Città	Testo	
Provincia	Testo	
Data di nascita	Data/ora	
e-mail	Testo	
Note	Memo	

Telefoni

Nome campo	Tipo dati	Nota
IDTelefoni	Numerazione automatica	Chiave primaria
IDStudente	Numerico	
Numero	Testo	
Tipo	Testo	

Certificazioni

Nome campo	Tipo dati	Nota
Titolo	Testo	Chiave primaria
Categoria	Testo	
Descrizione	Memo	

Dettaglio certificazioni

Nome campo	Tipo dati	Nota
IDCertificazione	Numerazione automatica	Chiave primaria
IDStudente	Numerico	
Titolo	Testo	
Data Conseguimento	Data/ora	

Docenti

Nome campo	Tipo dati	Nota
IDDocente	Numerazione automatica	Chiave primaria
Nome	Testo	
Cognome	Testo	
Indirizzo	Testo	
CAP	Testo	
Città	Testo	
Provincia	Testo	
Data di nascita	Data/ora	
Foto	Oggetto Ole	
Note	Memo	
E-mail	Testo	
Sito	Collegamento ipertestuale	
Dipendente	Si/No	

Segue

Seguito

Nome campo	Tipo dati	Nota
Data di assunzione	Data/ora	
Codice Fiscale	Testo	
Partita IVA	Testo	Se non fosse un testo un eventuale zero iniziale verrebbe eliminato
Codice IBAN	Testo	

Materie

Nome campo	Tipo dati	Nota
Materia	Testo	Chiave primaria
Descrizione	Memo	
IDDocente	Numerico	

Valutazioni finali

Nome campo	Tipo dati	Nota
IDStudente	Numerico	Chiave primaria
Materia	Testo	
Voto	Numerico	

Nel database sono già presenti alcune maschere, che permettono l'immissione dei dati, in particolare una maschera per immettere l'elenco delle materie disponibili. Il docente può essere scelto con una casella combinata che mostra i nomi dei docenti della scuola (Figura 0.7).

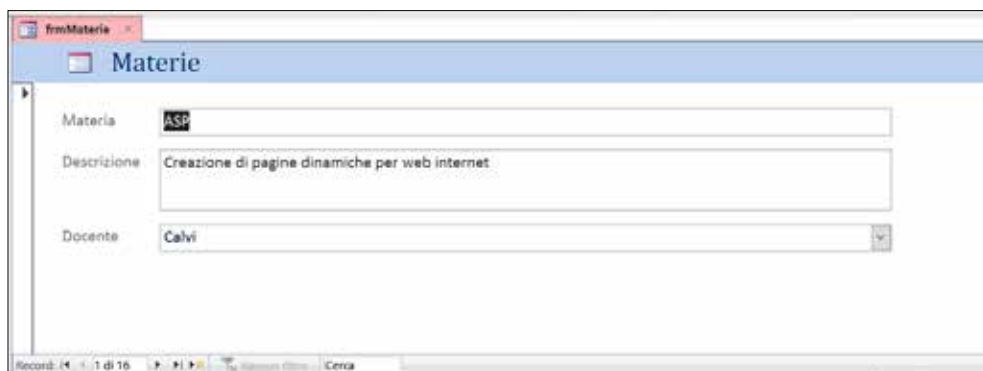


Figura 0.7 - La maschera frmMaterie.

Sono già pronte anche la maschera *frmTelefoni* per immettere i numeri di telefono degli studenti, e la maschera *frmCertificazioni* per immettere l'elenco delle certificazioni che costituiscono un titolo di merito riconosciuto per gli studenti della scuola.

Abbiamo già preparato anche la maschera *frmDettaglioCertificazioni*, per immettere le certificazioni conseguite da ciascuno studente (Figura 0.8) e la maschera *frmValutazioniFinali* con il voto finale di ciascuno studente in ciascuna materia.



Figura 0.8 - La maschera *frmDettaglioCertificazioni*.

Un'applicazione database

Il database *Scuola_stella*, alla fine del libro, non sarà più un semplice database, ma diventerà una **Applicazione database**. Si tratta di qualcosa di più complesso di un semplice database, infatti costruiremo grazie a VBA una interfaccia utente personalizzata dedicata a risolvere le esigenze specifiche degli utenti.

Autore e utente

Nel libro parleremo di utenti dei file realizzati con VBA, distinguendolo da chi ha creato il file. Nella realtà, molto spesso, le due figure coincideranno e voi scriverete i vostri progetti VBA per vostro uso personale. La distinzione fra utente e autore dei progetti è comunque una distinzione di ruoli.

Quando create i vostri progetti, siete autori, quando li usate, diventate utenti.

Quando siete autori, dovete pensare a creare un progetto semplice da usare e che non presenti problemi per l'utente. Anche se l'unico vostro utente siete voi stessi, non è certo simpatico ritrovarsi con un progetto che funziona male e che, durante l'uso, deve essere sistemato.

Se poi i vostri utenti non siete voi stessi, tanta maggiore attenzione dovrete prestare alla correttezza del vostro lavoro e alla semplicità d'uso della applicazione database che state creando. Tutto ciò è ancora più importante se pensate che chi userà la vostra applicazione non è molto esperto di database: cercate di rendere il suo lavoro più semplice e facile possibile.