



Su www.hoepliscuola.it è presente il link di accesso a esercitazioni per ogni sezione del Modulo 8 e simulazioni della prova d'esame con il simulatore Maxisoft.



8

USING DATABASES (DB)

- 1 Comprendere i database
- 2 Utilizzo dell'applicazione
- 3 Tabelle
- 4 Cercare informazioni
- 5 Oggetti
- 6 Stampa

SCOPI DEL MODULO

Il presente modulo richiede che il candidato comprenda il concetto di base dati (database) e dimostri di possedere competenza nel suo utilizzo.

Il candidato deve essere in grado di:

- comprendere cosa è un database, come è organizzato e come opera;
- creare un semplice database e visualizzarne il contenuto in modi diversi;
- creare una tabella, definire e modificare campi e loro proprietà; inserire e modificare dati in una tabella;
- ordinare e filtrare una tabella o una maschera; creare, modificare ed eseguire delle query per ottenere informazioni specifiche da un database;
- comprendere cosa è una maschera e crearne una per inserire, modificare ed eliminare record e dati contenuti nei record;
- creare dei report semplici e preparare delle stampe pronte per la distribuzione.



ECDL

Il seguente Syllabus fornisce i fondamenti per sostenere il test di tipo teorico e pratico relativo a questo modulo.

SEZIONE	TEMA	ARGOMENTO	PAG
1 Comprendere i database	1.1 Concetti fondamentali	1.1.1 Comprendere cosa è un database.	4
		1.1.2 Comprendere la differenza tra dati e informazioni.	4
		1.1.3 Comprendere come è organizzato un database in termini di tabelle, record e campi.	5
		1.1.4 Conoscere alcuni degli utilizzi più comuni di database di grandi dimensioni, quali: sistemi di prenotazione aerea, dati della Pubblica Amministrazione, dati di conti correnti, dettagli di pazienti in ospedale.	6
	1.2 Organizzazione di un database	1.2.1 Comprendere che ciascuna tabella di un database dovrebbe contenere dati relativi ad un solo argomento.	7
		1.2.2 Comprendere che ciascun campo di una tabella dovrebbe contenere un solo dato.	8
		1.2.3 Comprendere che il contenuto di un campo è associato ad un tipo di dato adeguato, quale: testo, numero, data/ora, sì/no.	8
		1.2.4 Comprendere che ai campi sono associate delle proprietà quali: dimensione, formato, valore predefinito.	8
		1.2.5 Sapere cosa è una chiave primaria.	9
		1.2.6 Comprendere cosa è un indice. Capire come permette di accedere più rapidamente ai dati.	10
	1.3 Relazioni	1.3.1 Comprendere che il motivo principale per creare relazioni tra tabelle di un database è minimizzare la duplicazione dei dati.	10
		1.3.2 Comprendere che una relazione viene costruita legando un campo univoco di una tabella ad un campo di un'altra tabella.	11
		1.3.3 Comprendere l'importanza di mantenere l'integrità delle relazioni tra tabelle.	12
	1.4 Operatività	1.4.1 Sapere che i database professionali sono progettati e creati da specialisti di database.	13
		1.4.2 Sapere che l'inserimento di dati, la gestione dei dati e il recupero delle informazioni vengono effettuati dagli utenti.	14
		1.4.3 Sapere che un amministratore di database fornisce accesso a dati specifici agli utenti opportuni.	14
1.4.4 Sapere che l'amministratore del database è il responsabile del recupero di un database dopo guasti o errori gravi.		15	
2 Utilizzo dell'applicazione	2.1 Lavorare con i database	2.1.1 Aprire, chiudere un'applicazione di database.	17
		2.1.2 Aprire, chiudere un database.	18
		2.1.3 Creare un nuovo database e salvarlo all'interno di un'unità disco.	19
		2.1.4 Mostrare, nascondere le barre degli strumenti. Minimizzare, ripristinare la barra multifunzione (ove disponibile).	21
		2.1.5 Usare la funzione di Guida in linea (help) del programma.	21
	2.2 Operazioni comuni	2.2.1 Aprire, salvare e chiudere una tabella, una query, una maschera, un report.	22
		2.2.2 Cambiare modalità di visualizzazione di una tabella, query, maschera, report.	24
		2.2.3 Eliminare una tabella, una query, una maschera, un report.	26
		2.2.4 Navigare tra i record di una tabella, di una query, di una maschera.	27
		2.2.5 Ordinare i record di una tabella, maschera, risultato di una query in ordine numerico crescente o decrescente, o in ordine alfabetico crescente o decrescente.	27
3 Tabelle	3.1 Record	3.1.1 Inserire, eliminare record in una tabella.	29
		3.1.2 Inserire, modificare, eliminare dati in un record.	31
	3.2 Progettazione	3.2.1 Creare e assegnare un nome a una tabella e specificarne i campi con i relativi tipi di dati, quali: testo, numero, data/ora, sì/no.	32
		3.2.2 Applicare le proprietà dei campi, quali: dimensioni del campo, formato numerico, formato data/ora, valore predefinito.	35
		3.2.3 Creare una regola di validazione per numeri, data/ora, valuta.	36
		3.2.4 Comprendere le conseguenze quando si modificano i tipi di dati, le proprietà dei campi di una tabella.	37

SEZIONE	TEMA	ARGOMENTO	PAG
		3.2.5 Definire un campo come chiave primaria.	38
		3.2.6 Indicizzare un campo con, senza duplicati.	38
		3.2.7 Inserire un campo in una tabella esistente.	39
		3.2.8 Modificare l'ampiezza delle colonne di una tabella.	40
4 Cercare informazioni	4.1 Operazioni fondamentali	4.1.1 Usare il comando di ricerca per trovare una specifica parola, numero, data in un campo.	43
		4.1.2 Applicare un filtro ad una tabella, maschera.	44
		4.1.3 Eliminare l'applicazione di un filtro da una tabella, maschera.	47
	4.2 Query	4.2.1 Comprendere che una query viene utilizzata per estrarre e analizzare dei dati.	48
		4.2.2 Creare e salvare una query su una sola tabella usando specifici criteri di ricerca.	48
		4.2.3 Creare e salvare una query su due tabelle usando specifici criteri di ricerca.	50
		4.2.4 Aggiungere criteri ad una query utilizzando uno o più dei seguenti operatori: = (uguale), <> (diverso), < (minore di), <= (minore di o uguale a), > (maggiore di), >= (maggiore di o uguale a).	52
		4.2.5 Aggiungere criteri ad una query utilizzando uno o più dei seguenti operatori logici: AND, OR, NOT.	53
		4.2.6 Utilizzare un carattere jolly in una query: *, %, ? oppure _.	54
		4.2.7 Modificare una query: aggiungere, modificare, eliminare dei criteri.	56
		4.2.8 Modificare una query: aggiungere, eliminare, spostare, nascondere, visualizzare dei campi.	56
		4.2.9 Eseguire una query.	58
		5 Oggetti	5.1 Maschere
5.1.2 Creare e assegnare un nome ad una maschera.	60		
5.1.3 Utilizzare una maschera per inserire nuovi record.	63		
5.1.4 Utilizzare una maschera per eliminare record.	64		
5.1.5 Utilizzare una maschera per inserire, modificare, eliminare dati in un record.	65		
5.1.6 Inserire, modificare testo nell'intestazione, piè di pagina di una maschera.	65		
6 Stampa	6.1 Report, esportazione di dati		
		6.1.2 Creare e assegnare un nome ad un report basato su una tabella, una query.	70
		6.1.3 Modificare la collocazione dei campi di dati e le intestazioni all'interno di un report.	75
		6.1.4 Presentare campi specifici in un report raggruppati per somma, minimo, massimo, media, conteggio, con punti di interruzione adeguati.	77
		6.1.5 Inserire, modificare il testo nell'intestazione, piè di pagina di un report.	79
		6.1.6 Esportare una tabella, il risultato di una query in formato foglio elettronico, testo (.txt, .csv), XML all'interno di un'unità disco.	81
		6.2 Stampa	6.2.1 Modificare l'orientamento (verticale, orizzontale) di una tabella, maschera, risultato di una query, report. Modificare le dimensioni della carta.
	6.2.2 Stampare una pagina, record selezionato/selezionati, tabella completa.		85
	6.2.3 Stampare tutti i record, o pagine specificate, usando la visualizzazione maschera.		87
	6.2.4 Stampare il risultato di una query.		88
	6.2.5 Stampare una o più pagine specifiche di un report, un report completo.		88

IMPORTANTE! Durante ciascuna prova d'esame, **il candidato non deve mai utilizzare alcuna combinazione di tasti** (scorciatoie in alternativa all'uso del mouse), in particolare le combinazioni: **Ctrl + Alt + Canc, Alt + F4, Shift + Tab, Alt + Tab, Ctrl + C, Ctrl + X, Ctrl + V**. **Non è ammesso neppure l'uso dei tasti funzione**, per esempio **F1** per consultare l'Help in linea.

1 COMPRENDERE I DATABASE



Qualsiasi **insieme strutturato di informazioni**, come per esempio un dizionario, oppure l'elenco dei DVD di una videoteca, può essere considerato un **database** (base di dati o archivio) e il sistema software che ne permette la realizzazione è definito DBMS (DataBase Management System), come per esempio **Microsoft Office Access 2010** (che, per semplificare, chiameremo Access), applicativo del pacchetto **Office 2010**.

1.1 CONCETTI FONDAMENTALI

1.1.1 Comprendere cosa è un database



CLIPPY

◀ Un **database** si dice **relazionale** quando è strutturato su **più tabelle**, ognuna relativa a uno specifico argomento, tra loro correlate. ▶



Un ◀ **database relazionale** ▶ è un **insieme gestito di dati**, ovvero un archivio elettronico, organizzato in tabelle, che consente la gestione dei dati stessi in termini di inserimento, ricerca, modifica, consultazione, aggiornamento ecc.

Nella vita quotidiana spesso si utilizzano grandi archivi, talvolta cartacei, come la rubrica telefonica, altre volte elettronici, come l'elenco dei libri di una biblioteca disponibili online.

Il **vantaggio** più **immediato** di un **archivio elettronico** è dato, per esempio, dalla possibilità di **ordinare** e di **filtrare** i dati in base a diversi criteri. Per esempio, se di un libro si ricorda solo l'autore, inserendolo come criterio di ordinamento si otterrà la lista dei libri ordinati per autore; se si conosce il titolo, si chiederà di visualizzare l'elenco dei libri in ordine alfabetico per titolo; se si desidera visualizzare solo i libri di uno specifico genere, si potrà applicare un filtro in base a questo parametro e così via.

1.1.2 Comprendere la differenza tra dati e informazioni



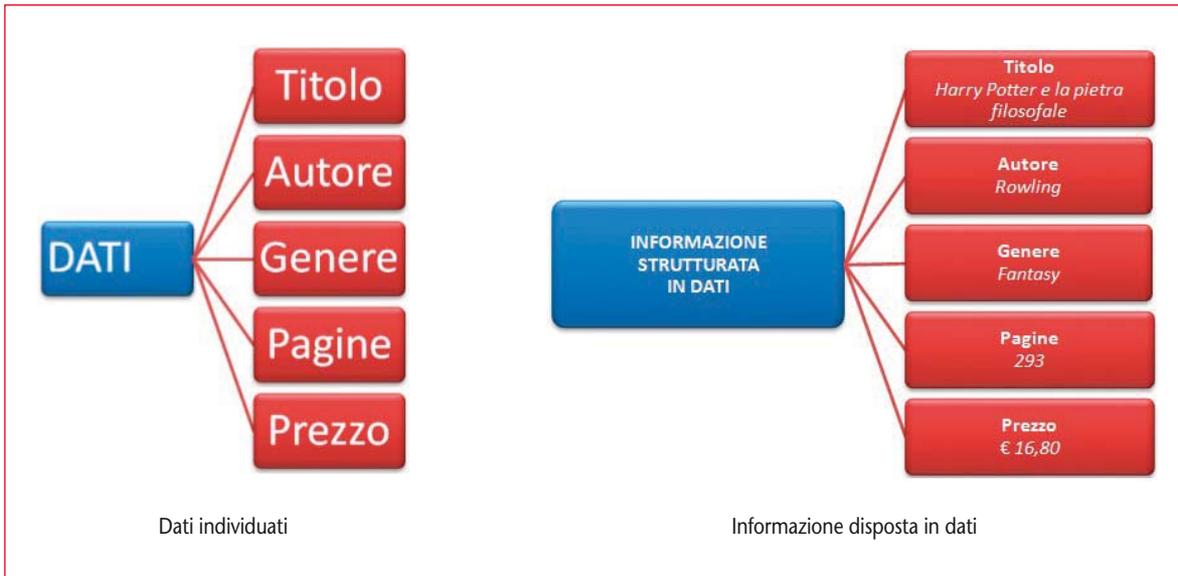
In un database il termine **dato** indica una **descrizione elementare**, talvolta espressa anche mediante un codice, che, **strutturata con altre**, è in grado di rappresentare un'**informazione**.

Da ciò si comprende che, per poter essere gestite da un database, le informazioni devono essere organizzate in dati. Proponiamo un esempio di una informazione che struttureremo in dati.

INFORMAZIONE

Il libro *Harry Potter e la pietra filosofale*, dell'**autrice** Rowling, è di **genere** Fantasy, si compone di 293 **pagine** e il suo **prezzo** è di € 16,80.

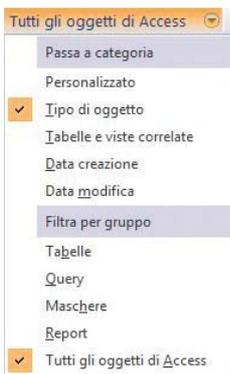
ORGANIZZAZIONE DELL'INFORMAZIONE IN DATI



1.1.3 Comprendere come è organizzato un database in termini di tabelle, record e campi



Un **database** è un **file** che può contenere diversi **oggetti**, ognuno dei quali preposto a una determinata funzione.



- **Tabelle:** costituiscono l'elemento primario di ogni database, ossia l'**archivio** in cui sono inseriti i dati relativi a uno specifico argomento.
- **Query:** sono interrogazioni che permettono di eseguire ricerche ed estrazioni sui dati memorizzati nelle tabelle in base a specifici criteri definiti dall'utente.
- **Maschere:** attraverso un'interfaccia grafica, consentono di aggiungere, modificare, eliminare dati nelle corrispondenti tabelle.
- **Report:** permettono di stampare i dati inseriti nelle tabelle e quelli ottenuti dalle interrogazioni, con layout personalizzati.

Nella tabella i dati sono organizzati in **campi**, visualizzati in colonne, e in **record**, costituiti da righe.

		Campi					
		Codice libro	Titolo	Genere	Prezzo di copertina	Numero di pagine	Codice autore
Record	LA001	Harry Potter e la pietra filosofale	Fantasy	€ 16,80	293	AA001	
	LA002	Dieci piccoli indiani	Giallo	€ 7,80	210	AA002	
	LA003	Harry Potter e il prigioniero di Azkaban	Fantasy	€ 14,50	366	AA001	
	LA004	Io non ho paura	Romanzo	€ 9,50	219	AA003	
	LA005	Le stagioni della vita	Filosofico	€ 7,40	164	AA004	



Campo: è ogni **singolo dato** contenuto in una tabella, ovvero uno **specifico elemento** di informazione.

Per esempio, nella tabella **Libri** il campo **Genere** dà origine all'omonima colonna in cui sono inseriti tutti i generi dei testi presenti nel database.



Record: è un **insieme di campi** che definisce in modo **completo** l'**informazione** a cui si riferisce.

Per esempio, nella tabella **Libri**, in cui sono previsti i campi **Codice libro**, **Titolo**, **Genere**, **Prezzo di copertina**, **Numero di pagine**, **Codice autore**, l'inserimento dei relativi dati nei sei campi genera un record.

1.1.4 Conoscere alcuni degli utilizzi più comuni di database di grandi dimensioni, quali: sistemi di prenotazione aerea, dati della Pubblica Amministrazione, dati di conti correnti, dettagli di pazienti in ospedale

Quasi tutte le aziende private e le Pubbliche Amministrazioni si avvalgono di sistemi di memorizzazione ed elaborazione basati su **database relazionali**. Questi grandi archivi elettronici, infatti, risultano **adatti a gestire su larga scala** servizi di diversa tipologia, consentendo un immediato accesso agli stessi con risparmio di tempo e di personale preposto.

Forniamo qualche esempio.



- **Sistemi di prenotazione aerea:** i siti delle compagnie aeree offrono la possibilità di inserire la data e la destinazione desiderate e ottenere in tempo reale informazioni sulla disponibilità di posti e l'accesso alla prenotazione.
- **Dati della Pubblica Amministrazione:** i siti di numerosi Enti pubblici offrono diverse opportunità di utilizzo dei propri database. Per esempio, il PRA (Pubblico Registro Automobilistico) permette consultazioni per ottenere informazioni sullo stato giuridico attuale del veicolo; il Catasto consente visure per individuare i beni immobili posseduti da un soggetto; il Ministero dell'Economia e delle Finanze permette la compilazione della dichiarazione dei redditi (per esempio, del modello 730) e il suo invio telematico ecc. Anche gli archivi di Stato, incaricati della conservazione dei documenti prodotti dallo Stato e degli atti storici che interessano lo Stato stesso, offrono in alcuni casi la consultazione online.
- **Dati di conti correnti:** ogni utente che ha stipulato con un istituto di credito (o con Poste Italiane) un contratto di gestione online del conto corrente può accedere al database della banca, consultare il proprio conto, disporre bonifici, effettuare ricariche telefoniche, investire in azioni ecc. in piena autonomia.
- **Informazioni di pazienti ricoverati in ospedale:** le cartelle cliniche dei pazienti vengono "digitalizzate", ovvero archiviate in formato elettronico: in questo modo esse sono disponibili per la consultazione, per esempio mediante rete intranet aziendale, da parte dei primari, dei medici o dal personale infermieristico dei diversi reparti.

1.2 ORGANIZZAZIONE DI UN DATABASE

1.2.1 Comprendere che ciascuna tabella di un database dovrebbe contenere dati relativi ad un solo argomento

Il motivo per cui un database è in genere costituito da più tabelle, ognuna con dati relativi a uno specifico argomento, è per **evitare** la cosiddetta **ridondanza**, ovvero la ripetizione di informazioni. Per esempio, se si considera un semplice database che cataloga i libri di una biblioteca, sarebbe sbagliato produrre una sola tabella perché se di un autore esistono più libri si dovrebbe, per ognuno, indicare ogni volta tutti i dati dell'autore producendo così inutili duplicazioni.

Codice libro	Titolo	Genere	Prezzo di copertina	Numero di pagine	Cognome	Nome	Nazionalità	Anno di nascita
LA001	<i>Harry Potter e la pietra filosofale</i>	Fantasy	€ 16,80	293	Rowling	Joanne Kathleen	Britannica	1965
LA002	<i>Dieci piccoli indiani</i>	Giallo	€ 7,80	210	Christie	Agatha	Britannica	1890
LA003	<i>Harry Potter e il prigioniero di Azkaban</i>	Fantasy	€ 14,50	366	Rowling	Joanne Kathleen	Britannica	1965
LA004	<i>Io non ho paura</i>	Romanzo	€ 9,50	219	Ammanniti	Niccolò	Italiana	1966
LA005	<i>Le stagioni della vita</i>	Filosofico	€ 7,40	164	Hesse	Hermann	Tedesca	1877

Invece, per **strutturare in modo corretto** le informazioni, si potrebbe pensare a due tabelle, una con i dati relativi ai libri e un'altra con i dati degli autori; in entrambe sarà inserito un campo mediante il quale **istituire il collegamento** che renderà possibile ottenere per ogni libro anche tutte le informazioni relative all'autore.

Campi

TABELLA LIBRI

Codice libro	Titolo	Genere	Prezzo di copertina	Numero di pagine	Codice autore
LA001	<i>Harry Potter e la pietra filosofale</i>	Fantasy	€ 16,80	293	AA001
LA002	<i>Dieci piccoli indiani</i>	Giallo	€ 7,80	210	AA002
LA003	<i>Harry Potter e il prigioniero di Azkaban</i>	Fantasy	€ 14,50	366	AA001
LA004	<i>Io non ho paura</i>	Romanzo	€ 9,50	219	AA003
LA005	<i>Le stagioni della vita</i>	Filosofico	€ 7,40	164	AA004

Record

TABELLA AUTORI

relazione istituita sul campo Codice autore

Codice autore	Cognome	Nome	Nazionalità	Anno di nascita
AA001	Rowling	Joanne Kathleen	Britannica	1965
AA002	Christie	Agatha	Britannica	1890
AA003	Ammanniti	Niccolò	Italiana	1966
AA004	Hesse	Hermann	Tedesca	1877

È evidente che la creazione di due tabelle distinte per argomento evita di ripetere, per ogni libro del medesimo autore, tutte le notizie a lui riferite.

1.2.2 Comprendere che ciascun campo di una tabella dovrebbe contenere un solo dato

Il **campo** rappresenta l'unità di base per l'immissione dei dati in un record e proprio per tale motivo **dovrà contenere un solo elemento di informazione**, una categoria specifica. Per esempio, il campo **Cognome** includerà solo i cognomi, il campo **Prezzo di copertina** avrà solo gli importi in euro, e così via.

1.2.3 Comprendere che il contenuto di un campo è associato ad un tipo di dato adeguato, quale: testo, numero, data/ora, sì/no

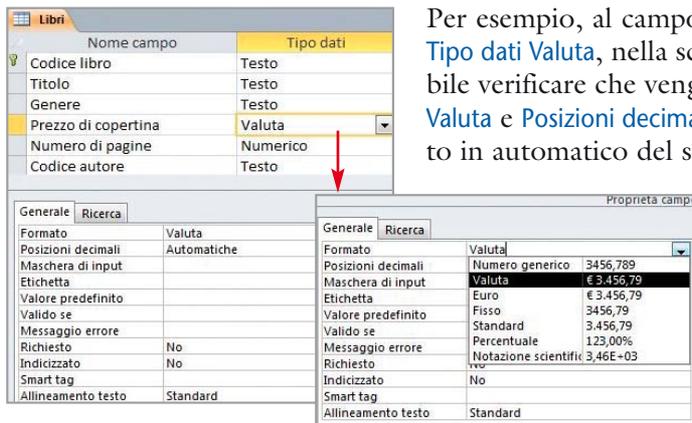
Durante la fase di creazione della tabella, oltre ad **attribuire a ogni campo il nome che lo identifica**, visualizzato nell'intestazione di colonna, è necessario **definire il tipo di dati** che potrà contenere. I dati inseriti in un campo devono soddisfare la caratteristica di **omogeneità**, di conseguenza essi devono avere **tutti la stessa natura**; ciò significa che lo stesso campo non potrà contenere, per esempio, dati di tipo testo e dati di tipo numerico.

Nella tabella che segue sono indicati i **tipi di dati** disponibili in Access, le loro caratteristiche e l'utilizzo che si può fare del contenuto.

TIPO DI DATO	CARATTERISTICHE E UTILIZZO DEL CONTENUTO
Testo	Testo (nome, cognome, città ecc.), insieme di testo e numeri (indirizzi, CF), o numeri non suscettibili di calcoli (numeri telefonici, CAP, P. IVA); può contenere fino a 255 caratteri.
Memo	Testo particolarmente lungo formato anche da numeri (descrizioni, note o commenti); può contenere fino a circa 1 GB di dati, ma le impostazioni predefinite visualizzano i primi 64.000 caratteri.
Numerico	Dati numerici, che possono essere oggetto di calcolo, la cui rappresentazione può avvenire in diversi modi: byte, intero, intero lungo, precisione singola, precisione doppia, ID replica, decimale.
Data/Ora	Cifre di data e ora con possibilità di eseguire calcoli.
Valuta	Cifre, precedute dal simbolo della valuta (€ ecc.), utilizzate per calcoli.
Numerazione automatica	Inserisce nel campo un numero sequenziale univoco a incremento automatico per ogni record aggiunto nella tabella (1, 2, 3 ecc.); non può essere aggiornato.
Si/No	Dato che può assumere uno solo dei due valori previsti (Si/No, Vero/Falso, On/Off).
Oggetto OLE (Object Linking Embedded)	Oggetto collegato o incorporato in una tabella (immagine, grafico, documento di Word, foglio di lavoro di Excel).
Collegamento ipertestuale	Indirizzo per collegarsi a un documento o a un file disponibile sul Web, in una LAN o nel computer locale.
Allegato	Allega al record immagini e altri tipi di file anche in formato compresso.
Calcolato	Consente di impostare calcoli tra campi della stessa tabella e ne mostra i risultati.
Ricerca guidata	Consente di aggiungere una caratteristica alla tabella prelevando dati da un'altra attraverso una casella di riepilogo combinata.

1.2.4 Comprendere che ai campi sono associate delle proprietà quali: dimensione, formato, valore predefinito

Definito il tipo di dati memorizzabili in un certo campo, il programma ne attribuisce automaticamente le **proprietà**, ovvero gli **attributi** che definiscono il modo in cui ciascun campo verrà memorizzato e gestito da Access. Tali proprietà, **che variano in relazione al tipo di dato scelto**, possono essere modificate dall'utente.



Per esempio, al campo **Prezzo di copertina**, dopo aver definito il **Tipo dati Valuta**, nella scheda **Generale** in **Proprietà campo**, è possibile verificare che vengono applicati come predefiniti il **Formato Valuta** e **Posizioni decimali Automatiche**; ciò comporta l’inserimento in automatico del simbolo dell’euro davanti alla cifra digitata, con l’aggiunta della virgola e di due posizioni decimali. Per modificare le proprietà predefinite, vedere il [punto 3.2.2](#).

Di seguito analizziamo le **proprietà** disponibili nella scheda **Generale** per un **campo** di tipo **Testo**.

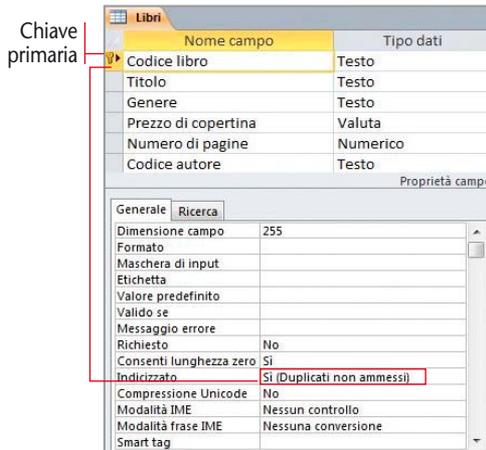
PROPRIETÀ	ATTRIBUTI DEL CONTENUTO
Dimensione campo	Permette di stabilire il numero di caratteri che si possono inserire nel campo.
Formato	Stabilisce la modalità di visualizzazione dei dati.
Maschera di input	Consente di specificare una maschera di immissione, visibile quando si inseriscono i dati nel campo.
Etichetta	Permette di inserire un diverso nome che si desidera assegnare al campo nelle maschere e nelle intestazioni dei report.
Valore predefinito	Si può digitare il dato che si vuole visualizzare in automatico nel campo, in ogni record.
Valido se	Consente di inserire regole di validazione, ovvero stabilire dei limiti ai dati immessi.
Messaggio errore	È possibile digitare il testo che il programma farà comparire nel caso di immissione di dati che non rispettino le regole di validazione.
Richiesto	Stabilisce se rendere o meno obbligatorio l’inserimento di dati nel campo.
Consenti lunghezza zero	Permette di considerare valide le immissioni di stringhe di testo a lunghezza zero in campi di tipo Testo , Memo e Collegamento ipertestuale .
Indicizzato	Consente di creare un indice rispetto al campo e di ammettere o negare l’inserimento di duplicati, ovvero di dati uguali in record diversi.
Compressione Unicode	Applicata ai campi di tipo Testo , Memo o Collegamento ipertestuale con qualche limitazione, tale proprietà comprime i dati memorizzati per espanderli al momento del recupero.
Modalità IME	In elaboratori con versione asiatica di Windows e Input Method Editor (IME) installato, controllano la conversione dei caratteri in set di caratteri di lingue asiatiche.
Modalità frase IME	
Smart tag	Indica il nome dello smart tag registrato e l’azione a cui si desidera associarlo.

1.2.5 Sapere cosa è una chiave primaria

Quando si consultano grandi archivi non è raro trovare **situazioni di omonimia**, di date di nascita coincidenti e, in generale, di dati duplicati **che possono rendere dubbia l’informazione reperita**.



Per evitare che ciò accada, in fase di strutturazione di una tabella è necessario che ogni record inserito possa essere individuato in maniera univoca, ovvero che includa almeno un campo in cui inserire **valori non replicabili**. Tale **campo** deve essere impostato come **Chiave primaria** per **identificare in modo univoco il record a cui è legato**.



Qualora un solo campo non riesca a soddisfare la condizione di non replicabilità, è possibile assegnare la chiave primaria a un set (combinazione) di campi.

Tuttavia, se non si riesce a individuare il campo a cui attribuire la chiave primaria, si può richiedere al programma l'inserimento di un campo **Numerazione automatica**, in grado di generare un numero sequenziale univoco, incrementato in automatico di un'unità ogni volta che viene aggiunto un nuovo record a una tabella. Questa operazione viene compiuta automaticamente da Access qualora, in fase di salvataggio di una nuova tabella, l'utente non indichi alcuna chiave primaria.

1.2.6 Comprendere cosa è un indice. Capire come permette di accedere più rapidamente ai dati



L'**indice** è un attributo che viene utilizzato dal programma per localizzare velocemente il campo a cui è associato.

Nel momento in cui si attribuisce la **Chiave primaria** a un campo, il programma lo **indicizza automaticamente** (non ammettendone duplicati), ovvero sulla base di tale campo crea un indice che viene utilizzato per operazioni di **ricerca** dei **dati**.

È possibile assegnare un **indice** anche ad altri campi in base ai quali si prevede di eseguire frequenti interrogazioni, in quanto tale attributo rende più **veloci** le operazioni di **ricerca**, **ordinamento** e **raggruppamento** sul campo indicizzato. Per esempio, considerando la **tabella Libri**, può accadere di eseguire spesso ricerche in base al **Titolo** del libro: se tale campo non viene indicizzato, Access dovrà scorrere tutti i record della tabella, ordinati in base al **Codice Libro** (in quanto **Chiave primaria**) e, in presenza di una tabella contenente migliaia di record, il risultato si farà attendere. Diversamente, se si indicizza il campo **Titolo**, Access creerà una tabella (non visibile) in cui tutti i record saranno ordinati in base al titolo e riuscirà quindi a fornire una risposta in brevissimo tempo.

L'**uso degli indici** deve essere ben considerato al momento della strutturazione del database, in quanto tale impostazione **dà origine a tante tabelle interne** quanti sono i campi indicizzati; ciò rende più complessa la struttura dell'archivio e richiede una maggiore quantità di memoria che, di conseguenza, **rallenta** le operazioni di **inserimento** di **nuovi record**.

1.3 RELAZIONI

1.3.1 Comprendere che il motivo principale per creare relazioni tra tabelle di un database è minimizzare la duplicazione dei dati

Al **punto 1.2.1** abbiamo visto come sia opportuno, per **evitare** la **duplicazione dei dati**, creare archivi organizzati su più tabelle, ognuna relativa a uno specifico argomento. Così facendo, nella tabella **Libri** sarà sufficiente prevedere un campo in cui inserire il **Codice autore** da utilizzare per collegare ogni libro alle informazioni di colui che lo ha

scritto, contenute nella tabella **Autori**: tale “collegamento” si realizza attraverso una **relazione**.

Creando relazioni è possibile:

- **visualizzare informazioni strutturate**, per cui ogni record di una tabella può mostrare tutti quelli correlati dell'altra;
- **eseguire query prelevando campi da più tabelle**, ottenendo come risposta record logici, ovvero insiemi di dati appartenenti a tabelle diverse.

Tabella **Autori** (primaria o padre) con visualizzati i record correlati della tabella **Libri** (secondaria o figlio)

Codice autore	Cognome	Nome	Nazionalità	Anno di nascita	Fare clic per aggiungere
AA001	Rowling	Joanne Kathleen	Britannica	1965	
AA002	Christie	Agatha	Britannica	1890	
AA003	Ammaniti	Niccolò	Italiana	1966	

Codice libro	Titolo	Genere	Prezzo di copertina	Numero di pagine
LA001	Harry Potter e la pietra filosofale	Fantasy	€ 16,80	293
LA003	Harry Potter e il prigioniero di Azkaban	Fantasy	€ 14,50	366
LA013	Harry Potter e la camera dei segreti	Fantasy	€ 16,80	307
LA015	Harry Potter e il calice di fuoco	Fantasy	€ 16,90	623
LA017	Harry Potter e l'ordine della fenice	Fantasy	€ 24,00	804
LA018	Harry Potter e il principe mezzosangue	Fantasy	€ 22,00	588
LA002	Dieci piccoli indiani	Giallo	€ 7,80	210
LA004	Io non ho paura	Romanzo	€ 9,50	219

Query che fornisce come risultato record logici, costituiti da campi prelevati dalle due tabelle correlate da una relazione.

Titolo	Genere	Cognome	Nome	Nazionalità
Harry Potter e il principe mezzosangue	Fantasy	Rowling	Joanne Kathleen	Britannica
Harry Potter e l'ordine della fenice	Fantasy	Rowling	Joanne Kathleen	Britannica
Harry Potter e il prigioniero di Azkaban	Fantasy	Rowling	Joanne Kathleen	Britannica
Harry Potter e il calice di fuoco	Fantasy	Rowling	Joanne Kathleen	Britannica
Harry Potter e la camera dei segreti	Fantasy	Rowling	Joanne Kathleen	Britannica
Harry Potter e la pietra filosofale	Fantasy	Rowling	Joanne Kathleen	Britannica
Le stagioni della vita	Filosofico	Hesse	Hermann	Tedesca
Dieci piccoli indiani	Giallo	Christie	Agatha	Britannica
La fabbrica di cioccolato	Humor	Dahl	Roald	Britannica
La forza della ragione	Romanzo	Fallaci	Oriana	Italiana
Io non ho paura	Romanzo	Ammaniti	Niccolò	Italiana
Un uomo	Romanzo	Fallaci	Oriana	Italiana
Tre metri sopra il cielo	Romanzo	Mocchia	Federico	Italiana
Ho voglia di te	Romanzo	Mocchia	Federico	Italiana
Scusa ma ti chiamo amore	Romanzo	Mocchia	Federico	Italiana
Il nome della rosa	Romanzo storico	Eco	Umberto	Italiana
L'opera al nero	Romanzo storico	Yourcenar	Marguerite	Belga
La spada e la luna	Romanzo storico	Pariani	Laura	Italiana
Viaggio alla fine del millennio	Romanzo storico	Yehoshua	Abraham	Israeliana
Baudolino	Romanzo storico	Eco	Umberto	Italiana

1.3.2 Comprendere che una relazione viene costruita legando un campo univoco di una tabella ad un campo di un'altra tabella

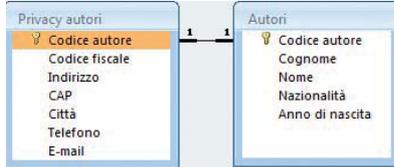
Le **relazioni** vengono di norma **impostate** tra un campo **Chiave primaria** di una tabella e un **campo** (denominato chiave esterna) che contiene lo **stesso tipo di dati** nella tabella da correlare.

Esistono **tre tipologie di relazioni** e per comprenderle meglio si propongono alcuni esempi.

■ Relazione Uno-a-uno



Nella relazione **Uno-a-uno** ogni record della prima tabella è correlato a un solo record della seconda, e viceversa.



Per esempio, se nel database **Biblioteca** si dovessero memorizzare informazioni private dei diversi autori, è possibile inserirle in una ulteriore tabella in cui ogni record corrisponderà a un solo autore.

■ Relazione Uno-a-molti



Nella relazione **Uno-a-molti** ogni record della prima tabella può essere correlato a più record della seconda.



Nell'esempio mostrato, tra la tabella **Autori** e la tabella **Libri** è stata impostata una relazione **Uno-a-molti** sul campo comune **Codice autore** in modo da ottenere, per qualsiasi autore rappresentato nell'omonima tabella, tutti i relativi libri contenuti nella tabella **Libri**.

■ Relazione Multi-a-molti



Nella relazione **Multi-a-molti** ogni record di una tabella è correlato a più record dell'altra e viceversa.



Per esempio, se nella biblioteca considerata si organizzano corsi di lettura di diverso tipo, è necessario costruire nel database tre ulteriori tabelle per gestire l'iscrizione ai corsi.

Considerato che ogni corso avrà più iscritti e che ciascun iscritto può frequentare più corsi, a ogni record della tabella **Corsi di lettura** corrisponderanno diversi record della tabella **Iscritti** e viceversa.

La relazione **Multi-a-molti** viene impostata scomponendola in due relazioni **Uno-a-molti** e creando una terza tabella (in questo caso, **Iscrizioni**), detta di collegamento, in cui saranno inseriti i campi **Chiave primaria** di ognuna delle due tabelle correlate.

La relazione **Multi-a-molti** viene impostata scomponendola in due relazioni **Uno-a-molti** e creando una terza tabella (in questo caso, **Iscrizioni**), detta di collegamento, in cui saranno inseriti i campi **Chiave primaria** di ognuna delle due tabelle correlate.

1.3.3 Comprendere l'importanza di mantenere l'integrità delle relazioni tra tabelle

Le **relazioni** tra tabelle devono essere **ben definite** e, per **evitare** situazioni di **ambiguità**, è necessario rispettare alcune condizioni:

- le tabelle devono trovarsi nello **stesso database**;
- nelle tabelle da correlare deve essere presente un **campo comune** (solitamente individuato dallo stesso nome);

- uno dei due campi delle due tabelle deve essere **chiave primaria**, o avere un indice che non ammette duplicati;
- i campi correlati devono contenere lo **stesso tipo di dati** (testo con testo, valuta con valuta ecc.), fatta eccezione per il campo **Numerazione automatica** che può essere **correlato anche** a un campo **Numerico** impostato a **Intero lungo**;
- i **campi** di tipo **numerico** correlati devono avere entrambi la **medesima Dimensione campo**.



L'**integrità referenziale** è un **sistema di regole** che si può **applicare alle relazioni** per assicurare che le stesse siano valide e per **evitare** che i **campi correlati** vengano **eliminati** o **modificati per errore**. Con questa funzione Access garantisce che ogni record della tabella correlata secondaria corrisponde a un record della tabella primaria.

L'**integrità referenziale** mette al riparo il database da possibili errori e impone precisi **vincoli** che impediscono:



- l'**aggiunta** di un record sul **lato molti** (relazione **Uno-a-molti**) senza l'esistenza di un record sul **lato uno** della relazione;
- l'**eliminazione** di un record sul **lato uno** (relazione **Uno-a-molti**) senza prima eliminare tutti i record corrispondenti sul **lato molti** della relazione;
- la **modifica** del valore di un **campo chiave primaria** di una tabella primaria dal quale dipendono i record di una tabella correlata.

Nell'immagine mostrata, nel database **Biblioteca** viene applicata l'integrità referenziale alla relazione **Uno-a-molti** tra le tabelle **Autori** e **Libri** in modo da attivare un controllo che impedisca:

- l'**aggiunta** di un libro nella tabella **Libri** se nella tabella **Autori** non è ancora stato inserito il relativo autore;
- la **cancellazione** di un autore dalla relativa tabella finché nella tabella **Libri** esistono suoi libri, ovvero record correlati;
- la **modifica** del **Codice autore** nella tabella **Autori** se esistono record correlati nella tabella **Libri**.

1.4 OPERATIVITÀ

1.4.1 Sapere che i database professionali sono progettati e creati da specialisti di database

I **database relazionali professionali** (si pensi, per esempio, a una base di dati che gestisce i voli aerei nazionali e internazionali) sono archivi complessi, di grandi dimensioni, che devono funzionare bene al fine di soddisfare le più svariate esigenze aziendali. Per questo motivo la loro **realizzazione** è affidata a esperti, con conoscenze tecniche molto specifiche, definiti **progettisti e sviluppatori**, che si occupano della progettazione di una base di dati e del delicato processo di corretto settaggio dei parametri di configurazione.



CLIPPY

◀ In questa fase il progettista dovrà prendere in considerazione anche le **esigenze degli utenti** del database, in termini di dati coinvolti nel processo e delle operazioni che è necessario eseguire su di essi. ▶



Il progettista genera la progettazione logica di un database, individua i ◀ **problemi** ▶ da risolvere e **analizza le** ◀ **esigenze** ▶ da soddisfare per consentire alle aziende di elaborare velocemente una maggiore quantità di dati al fine di controllare, pianificare e gestire in modo integrato le diverse attività.

Successivamente egli dovrà studiare i requisiti che il database dovrà avere in termini di proprietà e funzionalità, organizzare gli oggetti per l'archiviazione di dati, ovvero tabelle e relazioni, quindi stabilire le specifiche per gli oggetti logici, per esempio memorizzare procedure, pulsanti, funzioni ecc.



Lo **sviluppatore** si occupa di **codificare** e **testare** gli oggetti logici del database, perciò verifica il corretto funzionamento delle procedure di memorizzazione, dei pulsanti, delle funzioni ecc. Egli, inoltre, codifica e testa le applicazioni client-server.

Un database ben progettato potrà soddisfare esigenze di completezza, funzionalità, affidabilità, facilità d'uso con interfaccia intuitiva, quindi sarà idoneo e orientato a possibili sviluppi dimensionali dell'azienda.

1.4.2 Sapere che l'inserimento di dati, la gestione dei dati e il recupero delle informazioni vengono effettuati dagli utenti



CLIPPY

◀ Spesso nei **database professionali** gli utenti hanno a disposizione **strumenti**, quali semplici generatori di query e report, nonché interfacce grafiche in grado di rappresentare dati o sistemi di analisi dei dati. ▶

Generato il database, gli **amministratori di sistema** dovranno **istruire** gli utenti sulle **procedure** operative per un corretto utilizzo, illustrando gli ◀ **strumenti** ▶ a disposizione per effettuare elaborazioni e produrre informazioni appropriate.

L'**utente** quindi dovrà procedere con l'**inserimento dei dati** attraverso tabelle e maschere e sarà responsabile della loro corretta **gestione**. Inoltre, dovrà conoscere le tecniche per **estrapolare** in modo mirato le informazioni richieste. Per fare ciò dovrà essere in grado di impostare

query, ovvero digitare stringhe di parole chiave rappresentanti l'informazione richiesta, in modo da ottenere dal sistema un insieme di record che soddisfi le condizioni richieste.

Successivamente, mediante la realizzazione di report, l'utente potrà presentare su carta le informazioni ottenute.

1.4.3 Sapere che un amministratore di database fornisce accesso a dati specifici agli utenti opportuni



L'amministratore di database è un tecnico, definito anche sistemista, che si occupa dell'**installazione**, della **configurazione** e della **manutenzione** del database.

In particolare, egli **specifica le procedure operative** e risolve situazioni di eccezione non incluse nelle procedure pubblicate, prevede la pianificazione e l'esecuzione delle

operazioni quotidiane inerenti la disponibilità, il monitoraggio e l'ottimizzazione delle prestazioni, gli aggiornamenti e la risoluzione dei problemi.

Egli è in grado di provvedere alla configurazione del sistema e quindi può organizzare un **controllo** degli **accessi**, realizzato in genere attraverso un sistema di identificazione, in modo da **rendere disponibili alcuni dati solo a specifici utenti** affinché ne venga fatto un uso appropriato.

1.4.3 Sapere che l'amministratore del database è il responsabile del recupero di un database dopo guasti o errori gravi

Uno dei compiti più importanti dell'amministratore di un database è quello relativo alla **risoluzione dei problemi** causati da **errori procedurali** da parte degli utenti, oppure da **incidenti** dovuti a malfunzionamento di **hardware** o **software**.

L'**arresto** inaspettato dell'**applicazione** è la causa più comune di danneggiamento del database, quindi è importante assicurare la stabilità del sistema operativo e tenere in esecuzione solo le applicazioni necessarie per evitare conflitti software. Inoltre, se si lavora in zone soggette a interruzioni di alimentazione, l'amministratore dovrebbe prevedere l'utilizzo di un gruppo di continuità in grado di sopperire alla mancanza di energia elettrica.

Anche i problemi dei **dischi rigidi** (temperature troppo alte o basse, la presenza di campi elettrici e magnetici) potrebbero essere potenzialmente gravi, per cui tali dispositivi di memorizzazione non dovrebbero mai essere gli unici depositari dei dati.

Considerando tutto ciò, il modo migliore per **proteggere un database** contro tali eventi consiste nel pianificare una buona **strategia di backup** e, in generale, tecniche per **proteggere i dati**, quali:

- l'esecuzione dei backup in base a una programmazione ben definita;
- la memorizzazione dei backup in supporti di diversa tipologia (CR-ROM, DVD, hard disk esterni ecc.) da conservare in luoghi differenti;
- l'eventuale utilizzo di un **◀ gruppo di continuità ▶**;
- la verifica delle condizioni del disco fisso mediante appositi programmi di utilità;
- l'impiego di un software antivirus.



CLIPPY

◀ Il **gruppo di continuità** consiste in un apparecchio che ha al suo interno una o più batterie le quali immagazzinano corrente elettrica e, grazie a un sistema di conversione, la restituiscono al computer collegato in caso di mancanza di energia elettrica. ▶

M 8-1

1. Indica le affermazioni corrette.

- Un database relazionale è un insieme gestito di dati.
- Un database relazionale è un archivio elettronico organizzato su una tabella.
- In un database relazionale è possibile eseguire solo ordinamenti crescenti e decrescenti.
- In un database relazionale i dati possono essere ordinati in base a diversi criteri.

2. Indica le affermazioni false relativamente al concetto di dato.

- È sinonimo di informazione. Corrisponde a un campo.
- È una descrizione elementare. Corrisponde a un record.

3. Con il termine record si intende:

- una delle tabelle di cui si compone il database relazionale
- l'archivio del database
- l'oggetto che in un database è preposto alla stampa
- un insieme di campi

4. Indica quale dei seguenti non è un esempio di utilizzo su larga scala di database:

- archivi governativi
- prenotazione di voli online
- gestione informatizzata delle cartelle cliniche
- archivio dei dati aziendali

5. Indica il motivo per cui una tabella deve contenere dati relativi a un singolo oggetto.

- Perché il numero di colonne a disposizione è limitato.
- Per evitare la ridondanza di dati.
- Perché tutti i dati devono essere dello stesso tipo.
- Per favorire la ripetizione di dati.

6. È indispensabile indicare per ogni campo:

- il tipo di dati che dovrà contenere il formato
- le proprietà le dimensioni

7. La chiave primaria è:

- un campo che non conterrà valori ripetuti
- un campo indicizzato
- un campo che consente l'accesso alla tabella
- sinonimo di relazione

8. In una relazione l'integrità referenziale impedisce:

- l'eliminazione di record in entrambi i lati della relazione
- l'aggiunta di un record sul lato molti senza l'esistenza di un record sul lato uno
- l'aggiunta di un record sul lato uno della relazione
- l'eliminazione di un record sul lato molti della relazione

9. Considerando un database, l'amministratore ha la responsabilità di:

- estrapolarne i dati
- gestire le informazioni inserite
- inserirne i dati
- fornire la possibilità di accesso a dati specifici a particolari utenti



Puoi verificare la soluzione sul sito www.hoeplicuola.it dal link [Attività \(Attività\Soluzioni\M8\M8-1_Esercizio di consolidamento\)](#).