

Un **database** è una *raccolta di dati*.

Un database è composto di *tabelle*. Ogni tabella è composta da *record* e *campi*. I database possono essere composti da più tabelle.

Facciamo un esempio: una rubrica telefonica che raccoglie in un **Elenco** le *proprietà* considerate significative

Nome	Cognome	Indirizzo	CAP	Tel. abitazione
Paolo	Notini	via Pollenzo	16122	010-3346590
Maria	Filippi	via Zara	16145	010-4568928
Sara	Antoni	via Giusti	16121	010-5555555

I *campi* sono "Nome", "Cognome", "Indirizzo", "CAP", "Tel. abitazione" (ogni colonna) mentre i *record* sono gli utenti della rubrica cioè un insieme di campi (ogni riga).

Vediamo come si costruisce una rubrica telefonica con il database **Access** (uno dei tanti database presenti in commercio di tipo **Relational DataBase Management System**)

**Primo caso:** apriamo Access dal sistema operativo Windows:

Start – Programmi – Microsoft Access.

Scegliamo ora Nuovo – Database Vuoto.

Dobbiamo ora dare un nome significativo al nostro database (es. **rubrica**).

L'estensione del file rubrica è *.mdb* e in questo file verranno raccolti tutti i dati che andremo ad inserire.

Per ogni tabella, scegliamo la prima opzione del menù “Crea una tabella in visualizzazione **struttura**”. A questo punto dobbiamo creare la struttura della tabella **Elenco** inserendo i nomi dei campi che necessitano, il loro tipo ed eventualmente una descrizione facoltativa del campo stesso.

In Access è obbligatorio creare la “*chiave primaria*”: per default introduce un identificativo (ID) numerico.

## Chiave primaria

Ogni tabella deve contenere un campo che consenta di *identificare ogni dato in modo univoco*. Questo campo viene definito chiave primaria. Nell'esempio non possono esistere due o più IDElenco con lo stesso **valore**. Si potrà impostare questo campo come chiave primaria.

### Creazione di una chiave primaria:

1. Selezionare il campo con il tasto destro del mouse
2. Fare click sul pulsante **Chiave Primaria**

A questo punto sulla sinistra del campo compare una piccola chiave



Nome campo	Tipo dati	Descrizione
IDElenco	Contatore	
Nome	Testo	
Cognome	Testo	
Indirizzo	Testo	
CAP	Testo	
TelefonoAbitazione	Testo	

Nella **progettazione** di un database si parte dalla *definizione delle tabelle* che fanno parte del database. Per ogni tabella si *definiscono i campi* che rappresentano la **struttura** della tabella.

Poi si impostano le “*relazioni*” tra le tabelle che permettono di **normalizzare** (spezzare la *tabella grassa* in più tabelle *magre*) evitando **ridondanze**, raggiungendo un adeguato grado di **efficienza** e si prevederà un controllo su errori (**anomalie** di inserimento, cancellazione, aggiornamento) impostando l'**integrità referenziale**.

Supponiamo dunque di voler creare la rubrica telefonica di nostri **Amici** estraendo informazioni dalla tabella **Elenco** vista sopra evitando di creare “tabelle grasse” :

ID	Alias	email	CodElenco
1	Pupo	pupo@yahoo.it	1
2	Mami	mami@libero.it	3

i campi sono "Alias", "email" ed un campo (detto *chiave esterna*) che costituisce *riferimento* all'altra tabella "CodElenco" facendo attenzione che sia dello stesso tipo (Numerico) da cui poter estrarre ad esempio "Indirizzo", "CAP", "Tel. abitazione" senza ridondanze ed errori

Per ogni tabella, scegliamo la prima opzione del menù “Crea una tabella in visualizzazione **struttura**”.

Nome campo	Tipo dati	Descrizione
ID	Contatore	
Alias	Testo	
email	Testo	
CodElenco	Numerico	

A questo punto dobbiamo creare la struttura delle tabelle inserendo i nomi dei campi che necessitano, il loro tipo ed eventualmente una descrizione facoltativa del campo stesso.

## Tipi di dati e proprietà dei campi

Campo	Descrizione
<b>Testo</b>	Caratteri alfanumerici ( <i>nomi, indirizzi, cap, ...</i> ), max 255
<b>Memo</b>	Testo molto lungo con caratteri alfanumerici, max 65.535
<b>Numerico</b>	numeri su cui devono essere eseguiti calcoli
<b>Data e ora</b>	date e orari compresi tra 100 e 9999
<b>Valuta</b>	Utile per Euro e altre monete. Precisione 15 cifre + 4 dopo la virgola
<b>Contatore</b>	Numero progressivo univoco gestito da Access
<b>Sì/No</b>	Valori Sì / No oppure Vero / Falso
<b>Oggetto OLE</b>	Oggetto, Immagine collegato o incorporato in una tabella Access
<b>Collegamento Ipertestuale</b>	Collegamento a indirizzi URL
<b>Ricerca Guidata</b>	consente la creazione di un elenco dove è possibile selezionare un valore per ogni record.

Vediamo ora le proprietà più importanti descritte nella scheda Generale

Generale	Nome
<b>Dimensione Campo</b>	Lunghezza massima di un campo di testo o numerico. Limita la dimensione del database
<b>Formato</b>	Indica l'aspetto dei dati
<b>Etichetta</b>	Nome visualizzato come titolo della colonna. Utile perché il nome del campo normalmente non ha spazi bianchi e soprattutto per la creazione delle maschere e dei report
<b>Valore predefinito</b>	Valore inserito automaticamente ogni volta che viene creato un nuovo record
<b>Posizioni decimali</b>	Indica per i campi numerici e valuta il numero di cifre decimali
<b>Richiesto</b>	Se impostato su <b>Sì</b> obbligherà l'utente a inserire sempre un valore nel campo

Completata la struttura, chiudendo la finestra (icona “X”) chiede di salvare la tabella e con la scelta affermativa chiede il nome della tabella.

In Access si impostano le “relazioni” con facile interfaccia visuale e semplice selezione e trascinamento

e si imposta l’**integrità referenziale**



potendo gestire sia **anomalie** di cancellazione che di aggiornamento ed avviso di errore impedendo **anomalie** di inserimento,

Si distinguono infatti tre tipi di **anomalia**:

1. *Anomalia di inserimento.* Se nell’inserire un nuovo record in una tabella si è costretti a inserire informazioni già presenti nel DB.
2. *Anomalia di cancellazione.* Se nel cancellare un record si è costretti a cancellare informazioni che possono essere ancora utili nel DB.
3. *Anomalia di aggiornamento.* Se per aggiornare un record si è costretti ad aggiornarne molti altri.

Un errore di progettazione molto comune che dà quasi sempre luogo ad anomalie è quello di voler realizzare il Data-Base con un’unica grande tabella che contenga tutte le informazioni possibili. In gergo una tale tabella è detta **tabella grassa** (in inglese: *fat table*).

In Access la creazione di una *relazione* tra tabelle viene evidenziata da un segno + nella tabella principale (Elenco nell’esempio) a cui è correlata quella in cui è stata creata una *chiave esterna*

	IDElenco	Nome	Cognome	Indirizzo	CAP	Tel. abitazione
+	1	Paolo	Notini	via Pollenzo	16122	010-3346590
+	2	Maria	Filippi	via Zara	16145	010-4568928
+	3	Sara	Antoni	via Giusti	16121	010-5555555
*	(Contatore)					

## Verifica e modifica di una tabella

Prima di passare all’inserimento dei dati conviene verificare che la tabella soddisfi il nostro progetto. Apportare delle modifiche a una tabella con dati inseriti può portare spiacevoli conseguenze:

- Campo testo diventa campo numerico - Tutte le lettere saranno eliminate e rimarranno solo i numeri
- Riduzione della dimensione di un campo numerico - I dati che oltrepassano la nuova dimensione saranno tagliati.

## Modifica di una tabella

Conviene sempre *lavorare in visualizzazione struttura*. Per lavorare con una tabella chiusa basta farvi click sopra con il pulsante destro e scegliere “Visualizzazione Struttura”

## Inserimento di un campo

Per inserire un campo nuovo si può proseguire dopo l'ultimo campo presente in tabella. Se il nuovo campo deve essere inserito sopra a uno esistente basta fare:

1. Click sul campo esistente con il tasto destro del mouse
2. Scegliere **Inserisci righe**. La nuova riga verrà inserita sopra al campo esistente

## Eliminazione di un campo

Per eliminare un campo basta fare:

1. Click sul campo da eliminare con il tasto destro del mouse
2. Scegliere **Elimina righe**.
3. Dire di Sì per confermare l'eliminazione.

## Inserimento dei dati in una tabella

Una volta creata la struttura della tabella è possibile inserire i dati attraverso il menu "Visualizzazione" - "Foglio dati". Durante l'inserimento dei dati Access inserisce automaticamente le righe nuove. Per spostarsi da un campo all'altro è molto comodo utilizzare il tasto **TAB**. Se il campo chiave è di tipo contatore, verrà incrementato automaticamente.

## Salvare un record

Ogni nuovo record od ogni modifica viene automaticamente salvato da Access

## Eliminare un record

1. Selezionare il record da eliminare
2. Premere CANC
3. Confermare con Sì

I record eliminati **non** possono essere recuperati con il tasto annulla.

## LE QUERY

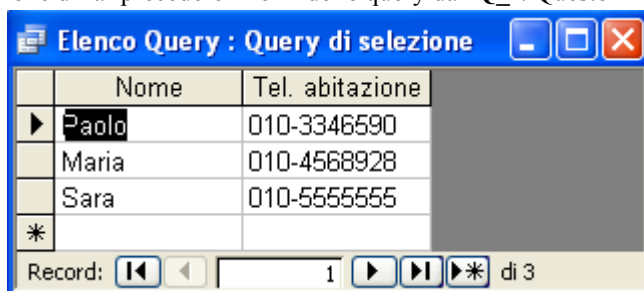
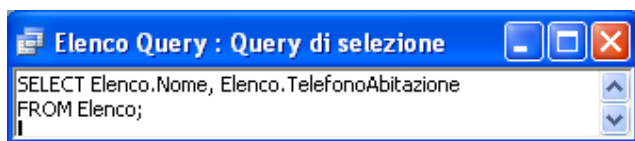
Una Query<sup>1</sup> è una visualizzazione dei dati contenuti su una o più tabelle, filtrati e/o aggregati secondo vari criteri. La traduzione letterale sarebbe "*interrogazione*", infatti la query è il risultato di una *domanda* posta al database. Esistono due tipi di query:

- **dettaglio** : vengono visualizzati tutti i campi di tutti i record
- **riepilogo**: consente di effettuare calcoli sui campi numerici (somma, media, minimo, massimo) oppure di scegliere raggruppamenti di date

Access semplifica la costruzione delle query mediante la creazione guidata.

## Creare una *query dettaglio* mediante una creazione guidata

1. Selezionare la tabella su cui si vuole costruire la maschera
2. Spostare sulla destra i campi che si vuole visualizzare. Il bottone ">" sposta un solo campo mentre il bottone ">>" li sposta tutti. Proseguire con Avanti.
3. Dare un nome alla query. Si può utilizzare la convenzione di far precedere i nomi delle query da "Q\_". Questo permette di distinguerle facilmente dalle tabelle.
4. A questo punto è possibile visualizzare il risultato dei dati raccolti.



<sup>1</sup> Il linguaggio SQL permette due tipi di query: interrogazione **statica** (tipo *compile and store*) e **dinamica** (tipo *compile and go*) costruita in modo interattivo (visuale) come stringa ed eseguita "on -fly".

- Con il tasto destro del mouse sulla query scegliere dal menu "Visualizzazione struttura".
- Modifichiamo la selezione da

```

SELECT Elenco.Nome, Elenco.TelefonoAbitazione
FROM Elenco;

in

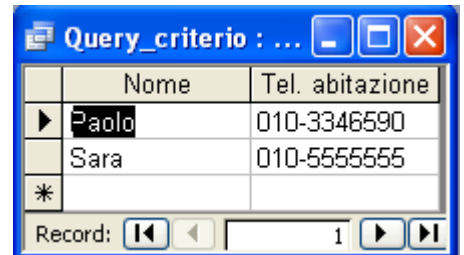
SELECT Elenco.Nome, Elenco.TelefonoAbitazione
FROM Elenco WHERE Nome > "p";

```



- Rieseguiamo la query chiudendo la finestra salvando le modifiche e selezionando dal menù principale "Query" –"Esegui". Notiamo che il risultato è differente perché in questo caso non visualizza i nomi prima della lettera "p".

In questo modo è possibile estrarre un sottoinsieme di dati secondo un preciso **criterio** deciso da noi: si può fare su un campo o su un insieme di campi legando le singole condizione con operatori logici "AND", "OR", "NOT".



- Eseguiamo ora una query di estrazione dati da entrambe le tabelle relazionate tra loro:



```

SELECT Elenco.Nome, Elenco.Cognome,
Amici.email

```

```

FROM Elenco INNER JOIN Amici ON
Elenco.IDElenco=Amici.CodElenco;

```

## Esercitiamoci

## Secondo caso: creiamo un DB di tipo Access da web-hosting Brinkster

Selezionare come User-Tool: “**Classic DataBase Manager**”



CLASSIC DATABASE MANAGER

Selezionare il pulsante “Create New Access 2000 DB”

Digitare nella finestra (aperta automaticamente) il nome significativo del database (es. **rubrica**) senza esplicitare l'estensione *.mdb* del file in cui verranno raccolti tutti i dati che andremo ad inserire, confermando con OK

Selezionare tra le opzioni del menu di scelta “DataBase” il nome scelto del database (automaticamente memorizzato con percorso **Root\db\nome.mdb** infatti la cartella *db* (con permesso di lettura e scrittura) conterrà i DB di tipo Access

**Inserire le query** desiderate nella “SQL<sup>2</sup> Box” (area di testo) utilizzando “*SQL ASSISTANT*” che mette a disposizione la possibilità di introdurre i comandi con corretta sintassi mediante il pulsante “Show Syntax” oppure di cancellare l'area di testo col pulsante “Clear SQL Box”.

Per **eseguire** ogni query sono previsti due pulsanti:

- “Execute Query & Get Results” esegue e visualizza la **tabella logica costruita al volo**, risultato dell'interrogazione
- “Execute Query, No Results Required” esegue senza visualizzare il risultato del comando

Supponiamo, selezionato il DB **rubrica**, di voler creare le tabelle della rubrica telefonica come abbiamo visto sopra.

- Scegliamo l'opzione dell'elenco “**Create table**”, premiamo il pulsante “Show Syntax” e personalizziamo il nome della tabella oppure editiamo direttamente il comando voluto nel SQL Box; in entrambi i casi, premiamo uno dei pulsanti per eseguire tale comando (nel caso di errori sintattici compariranno avvisi); infine cancelliamo l'area di testo col pulsante “Clear SQL Box” per inserire un comando alla volta.

**CREATE TABLE** Elenco  
(Nome varchar(20),  
Tel\_Abitazione char(11),  
IDElenco integer **primary key**)

**CREATE TABLE** Amici  
(Alias varchar(10),  
email varchar(20),  
CodElenco integer,  
ID integer **primary key**)

- A questo punto dobbiamo creare la struttura della nostra tabella inserendo i nomi dei campi che ci necessitano ed il loro tipo: scegliamo l'opzione “**Insert**”, personalizzando la sintassi proposta o editiamo direttamente il comando SQL

<b>INSERT INTO</b> Elenco (Nome, Tel_Abitazione, IDElenco) <b>VALUES</b> ('Paola Notini', '010-3346590', 1) .....	<b>INSERT INTO</b> Amici (Alias, email, CodElenco, ID) <b>VALUES</b> ('Pupo', 'pupo@yahoo.it', 1, 1) .....
--	---

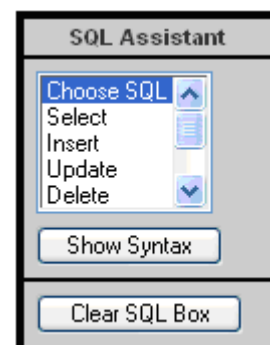
- Se vogliamo aggiornare con la modifica di un campo possiamo selezionare “**Update**”
- Se vogliamo aggiornare con cancellazione possiamo selezionare “**Delete**”
- Per compiere delle ricerche (**query di estrazione dati**) selezionare “**Select**”

**Select \* From** Amici

Query Results

Alias	email	CodElenco	ID
Pupo	pupo@yahoo.it	1	1
Mami	mami@libero.it	3	2

2 Record(s) Returned.



<sup>2</sup>Si utilizza la sintassi del linguaggio SQL (Structured Query Language) vedi <http://www.w3schools.com/sql/default.asp>

Select \* From Elenco

#### Query Results

Nome	Tel_Abitazione	IDElenco
Paola Notini	010-3346590	1
Maria Filippi	010- 45689	2
Sara Antoni	010-5555555	3

3 Record(s) Returned.

```
SELECT Elenco.Nome, Amici.email
FROM Elenco INNER JOIN Amici
ON Elenco.IDElenco=Amici.CodElenco
```

#### Query Results

Nome	email
Paola Notini	pupo@yahoo.it
Sara Antoni	mami@libero.it

2 Record(s) Returned.

Selezionando come [User-Tool](#) : “**DataBase Manager**”

i DB vengono automaticamente memorizzati con percorso



**Root\database\nome**

e si possono creare e gestire DB

non solo di il tipo Access



Date le prime nozioni sui database possiamo ora [utilizzare i database per ricerche effettuate via Web](#). Questa attività è sempre più utilizzata nel mondo Internet perché la maggior parte delle informazioni visualizzate nelle pagine WEB vengono raccolte nei database. Ecco i passi da effettuare per il *collegamento ad un database* con tecnologia ASP. Ricordiamo che il collegamento e le elaborazioni sui database non possono essere effettuate con linguaggio di marcatura HTML.

Per gestire i database con tecnologia ASP si utilizzano **strumenti ADO** (ActiveX Data Objects): un'architettura che fornisce oggetti di **alto livello** per **l'accesso universale ai dati** e in particolare contiene **oggetti per la comunicazione con i database** infatti ADO<sup>3</sup> consente di scrivere un'applicazione per la gestione e l'accesso ai dati contenuti in un server di database tramite un provider<sup>4</sup> OLE DB (interfaccia a basso livello che fornisce un modello di accesso ai dati universale che ne consente la gestione indipendentemente dal formato e dal metodo di memorizzazione; permette l'accesso non solo ai database relazionali ma a qualsiasi fonte dati : database **locali** o **remoti**, non relazionali, sistemi di file, posta elettronica, testo, grafica e oggetti multimediali, aziendali, personalizzati ...). Poiché tuttavia ciascun provider dispone di caratteristiche uniche, le modalità di interazione tra l'applicazione e ADO variano leggermente a seconda del provider stesso.

Per una descrizione più completa dell'architettura [ADO](#) si consulti:

#### [Il Modello ad Oggetti di ADO](#)

<http://www.soft-land.org/cgi-bin/doc.pl?doc=documenti/adomodel>  
(documento compresso scaricabile [Download il file Zip](#))

<sup>3</sup> Ideato come interfaccia dati client/server. Una limitazione di ADO consiste nell'impossibilità di creare fonti di dati ODBC pur se il provider predefinito è Microsoft OLE DB per ODBC (componente Microsoft Windows ed in particolare di Windows Open Services Architecture, che consente l'accesso a tutti i tipi di database relazionali).

<sup>4</sup> Il termine provider indica in generale un elemento fornitore dal quale si ottengono servizi o dati.