

Un **database** è una **raccolta di dati** (*strutturati e logicamente correlati*)

Un database è composto da *tabelle*. Ogni tabella è composta da *record* (**tuple**) e *campi*. I database possono essere composti da una o 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 o **tuple**).

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

Uso DBMS *tipo desktop*: 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* (versioni più datate) o *.accdb* (versioni più recenti) 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 é obbligatorio creare la “*chiave primaria*”: per default introduce un identificativo (ID) numerico.

Chiave primaria: dallo schema E-R allo schema logico

Nel derivare dallo schema E-R, ogni tabella deve contenere un *insieme di campi non nullo* che consenta di *identificare ogni tuple in modo univoco* (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 derivazione della schema E-R in *tabelle* che fanno parte del database. Per ogni tabella si *definiscono nomi e tipi dei campi* che rappresentano la **struttura** della tabella.

Poi si impostano le “*relazioni*” tra le tabelle che permettono di **normalizzare** (spezzare la *tabella grassa*, contenente tutte le informazioni, 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. Esiste un'**associazione** esprimibile come “**Amici**” in “**Elenco**”.

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 della tabella – come già illustrato – 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
Testo	Caratteri alfanumerici (<i>nomi, indirizzi, cap, ...</i>), max 255
Memo	Testo molto lungo con caratteri alfanumerici, max 65.535
Numerico	numeri su cui devono essere eseguiti calcoli
Data e ora	date e orari compresi tra 100 e 9999
Valuta	Utile per Euro e altre monete. Precisione 15 cifre + 4 dopo la virgola
Contatore	Numero progressivo univoco gestito da Access
Sì/No	Valori Sì / No oppure Vero / Falso
Oggetto OLE	Oggetto, Immagine collegato o incorporato in una tabella Access
Collegamento Iperestuale	Collegamento a indirizzi URL
Ricerca Guidata	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
Dimensione Campo	Lunghezza massima di un campo di testo o numerico. Limita la dimensione del database
Formato	Indica l'aspetto dei dati
Etichetta	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
Valore predefinito	Valore inserito automaticamente ogni volta che viene creato un nuovo record
Posizioni decimali	Indica per i campi numerici e valuta il numero di cifre decimali
Richiesto	Se impostato su Sì 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” (che traducono le **associazioni** tra entità) con facile interfaccia visuale e semplice selezione e trascinamento (da FK a PK).

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 e aggiornamento dei dati in una tabella

Una volta creata la struttura della tabella è possibile inserire o modificare 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

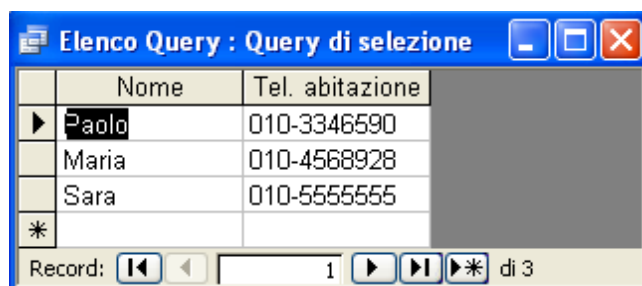
Una Query¹ è una visualizzazione dei dati contenuti su una o più tabelle, filtrati e/o aggregati secondo vari criteri. La traduzione letterale sarebbe "*interrogazione*", infatti la query è il risultato di una *domanda* posta al database. Esistono due tipi di query:

- **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.



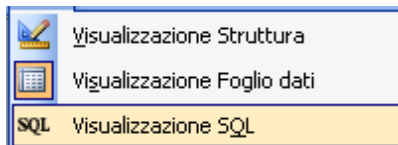
	Nome	Tel. abitazione	
▶	Paolo	010-3346590	
	Maria	010-4568928	
	Sara	010-5555555	
*			

Record: 1 di 3

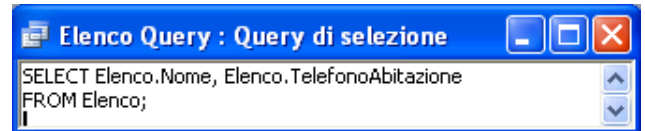
¹ Il linguaggio SQL permette due tipi di query: interrogazione **statica** (tipo *compile and store*) e **dinamica** (tipo *compile and go*) costruita in modo interattivo (visuale) come stringa ed eseguita “on -fly”.

Query in visualizzazione struttura e SQL

5. Con il tasto destro del mouse sulla query scegliere dal menu “Visualizzazione SQL”.

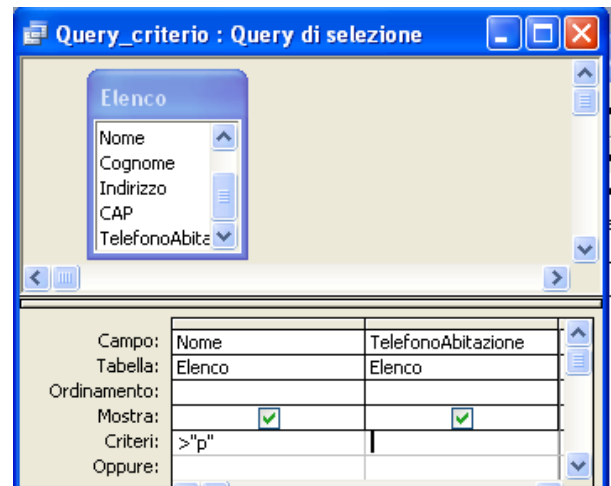


6. Modifichiamo la selezione da
SELECT Elenco.Nome, Elenco,TelefonoAbitazione
FROM Elenco;



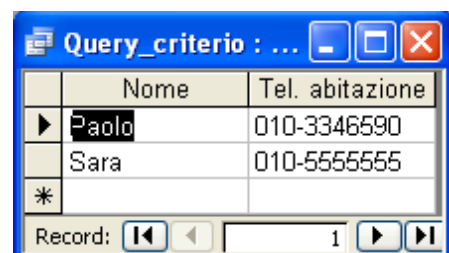
in

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



nb: risultato analogo se in “Visualizzazione Struttura” si imposta il criterio come in figura:

7. 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”.

SQL Statement	Syntax
AND / OR	SELECT column_name(s) FROM table_name WHERE condition AND OR condition

Query di estrazione da tabelle relazionate

8. 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;
```

equivalente a :

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

```
FROM Elenco, Amici
WHERE
Elenco.IDElenco=Amici.CodElenco;
```



[Esercitiamoci](#)

SQL – [Quick Reference](#) per le operazioni di *transazione*

SQL per inserimento

INSERT INTO	<p>INSERT INTO table_name VALUES (value1, value2, value3,...)</p> <p>or</p> <p>INSERT INTO table_name (column1, column2, column3,...) VALUES (value1, value2, value3,...)</p>
-------------	---

SQL per aggiornamento

UPDATE	<p>UPDATE table_name SET column1=value, column2=value,... WHERE some_column=some_value</p>
--------	---

SQL per cancellazione

DELETE	<p>DELETE FROM table_name WHERE some_column=some_value</p> <p>or</p> <p>DELETE FROM table_name (Note: Deletes the entire table!!)</p> <p>DELETE * FROM table_name (Note: Deletes the entire table!!)</p>
--------	--