

# Dispense di MS Access 2003



A cura di Paolo PAVAN - pavan@netlink.it

## Indice

Indice.....	1
Introduzione .....	2
Glossario .....	2
Cosa è un Database?.....	2
Requisiti di un DATABASE .....	3
La Normalizzazione dei Dati .....	3
Relazioni e Cardinalità in Access .....	6
Progettazione di un DatabaseE: un lungo cammino.....	6
DBMS e RDBMS .....	7
Dati e Matadati.....	8
Uso della Tastiera in Access .....	9
Creazione di un database.....	10
Interfaccia di Access .....	11
Analisi delle Tabelle database .....	12
Creazione del database.....	13
Definizione della Chiave Primaria .....	17
Creazione delle Relazioni .....	17
Uso dell'Autocomposizione ricerca per la definizione delle relazioni.....	19
Definizioni degli Indici .....	19
Filtri e Ordinamento dei dati.....	20
Formati di Access: conversioni dei database.....	21

**Attenzione!** Il presente manuale è in versione incompleta. Verrà terminato nei prossimi giorni.

## Introduzione

Vista la complessità dell'argomento, si affronteranno gli argomenti distinguendo una parte teorica introduttiva sulla logica e sulla progettazione dei database.

La parte più pratica vedrà in successivi passaggi la costruzione di un database relazionale prendendo spunto dal noto database Northwind.mdb incluso nella documentazione di Access.

L'esercizio nel suo complesso verte sulla costruzione di un database per la gestione di un negozio, che coinvolga i seguenti insiemi di dati:

- Categorie
- Prodotti
- Ordini
- Dettagli Ordini
- Clienti

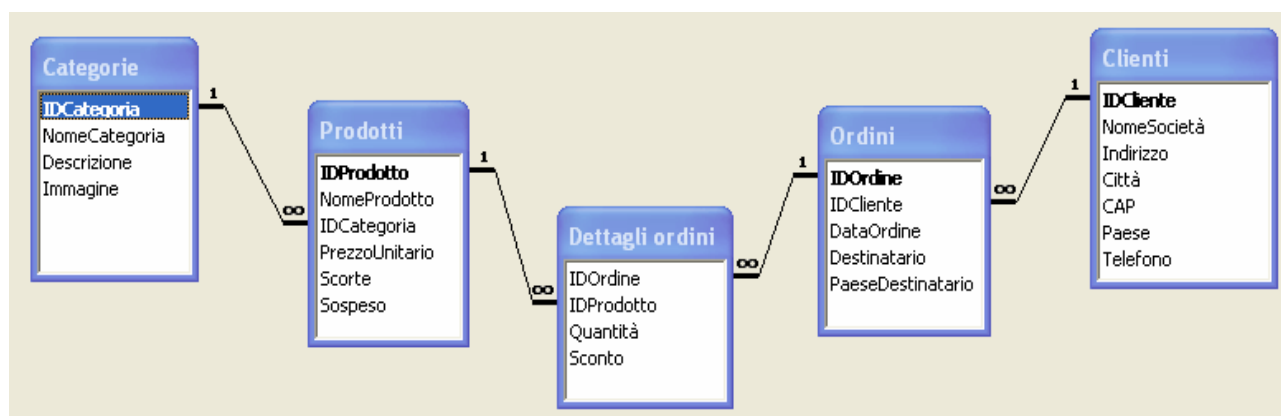
Si procederà con la creazione delle tabelle di base per poi procedere alla costruzione delle relazioni che tra di esse intercorrono.

La fase successiva prevede la creazione di maschere per agevolare l'inserimento e la visualizzazione dei dati.

L'uso delle Query consentirà poi di analizzare e interrogare i dati al fine di produrre "viste" dei dati consone alle nostre necessità.

L'uso finale dei Report consentirà di produrre versioni stampabili del contenuto del nostro archivio.

Per una migliore comprensione del manuale e della sua parte applicativa, pubblichiamo la struttura finale del nostro database di esempio:



## Glossario

1. DB = Database, base di dati
2. Archivio = sinonimo di database
3. DBMS = Database Management System (il sistema o programma con cui creiamo e gestiamo il DB)
4. RDBMS = Relational Database Management System
5. SQL = Structured Query Language, linguaggio per l'interrogazione e per la gestione dei dati
6. Query = operazione per l'interrogazione dei dati
7. Relazione = legame tra due elementi (campi) di una tabella

## Cosa è un Database?

Dal punto di vista teorico possiamo dire che un "database è un insieme di dati riferito ad un preciso sistema informativo".

Dal punto di vista pratico una base di dati, database o un archivio che dir si voglia, viene spesso erroneamente rappresentato come una semplice tabella.

Cognome	Nome	Telefono
Rossi	Mario	7878978
Verdi	Giuseppe	564654564
Bianchi	Luigi	54546
Rossi	Anna	7454564
Mazzini	Giuseppe	654456
Mazzola	Valentino	5454465

Invece un Database è formato da molte tabelle singole ma che possono essere in relazione tra loro, ovvero i dati di una tabella possono riguardare anche i dati delle altre.

Access consente quindi di gestire le relazioni, i legami tra i dati cosa che Excel non consente, agendo su volumi di dati e con politiche che un foglio elettronico non potrebbe mai fare.

## Requisiti di un DATABASE

Con il termine database si indica in genere un insieme di dati rivolti alla rappresentazione di un preciso sistema informativo di vario tipo, aziendale, scientifico, amministrativo o altro.

Il database rappresenta una **struttura** ben precisa dei dati, descritti come aggregazione di elementi base e classificati in modo specifico, in virtù della struttura comune che possiedono.

Il secondo elemento essenziale di un database è rappresentato dalle relazioni, tra i vari dati. Infatti sono importanti le informazioni strutturali di ogni categoria (tabella) del database ma anche le **relazioni** logiche esistenti tra le varie categorie. Un database correttamente strutturato deve rispondere ad una serie di quesiti:

- ridondanza minima: i dati non devono essere duplicati
- multiutenza: la base dati deve essere unica ma consultabile contemporaneamente da più utenti. In parole povere *le informazioni devono essere integrate ma distribuite*.
- permanenza dei dati. I dati devono essere fissi e duplicati su memorie fisse (hard disk o stream tape)

La realizzazione di una base dati completa avviene utilizzando un **DBMS** ovvero un database Management System, ovvero un programma (software) che permette la gestione di molteplici database che soddisfano i requisiti sopra citati.

## La Normalizzazione dei Dati

### Premessa

Una volta impostato lo schema Logico del nostro Database Relazionale è necessario effettuare una **serie di verifiche** sulle correttezza del procedimento svolto.

Questa operazione potrà causare delle modifiche dello schema stesso, in modo da renderlo corretto ed evitare il verificarsi, nella successiva gestione dei dati, di errori difficilmente ovviabili a posteriori.

Questo processo è detto **normalizzazione** e senza voler entrare nei dettagli della sua complessa teoria, possiamo analizzare le sue **regole principali** che vanno applicate rigorosamente a qualsiasi schema da noi impostato prima di procedere alla effettiva costruzione del database.

Il processo in questione è chiamato *normalizzazione dello schema relazionale* ed è basato su principi algebrici complessi, che però possono essere tradotti attraverso tre *forme* normali:

1. La *prima forma normale* stabilisce che in una tabella non possano esistere colonne che possano contenere più valori, creando una sorta di sottoinsiemi dei campi singoli. Si ovvia al problema creando per ciascun record della tabella di partenza tanti record quanti sono i valori multipli presenti nella colonna desiderata.

2. La *seconda forma normale* si riferisce alle tabelle in cui la chiave primaria non sia singola ma composta da più campi. In questo caso si stabilisce che tutte le colonne corrispondenti ai campi, siano strettamente dipendenti dalla chiave primaria.
3. La *terza forma normale* stabilisce che non possono esistere dipendenze (relazioni) tra colonne se non basate sulla chiave primaria, elemento essenziale per relazionare in modo corretto due elementi (campi) di una tabella.

Per comprendere il significato di questa operazione è necessario fare degli esempi pratici, in modo da capire cosa effettivamente comporta questo intervento. Vediamo di seguito alcuni casi in cui è stato possibile normalizzare i dati.

### La Prima Forma Normale

*Stabilisce che in una tabella non possono esistere colonne definite per contenere una molteplicità di valori.*

#### **Esempio**

N. Protocollo

Uffici Interessati

#### **Soluzione**

Quindi ogni riga deve contenere un solo valore.

#### **Esempio grafico**

Numero Protocollo	Uffici Interessati			

Numero Protocollo	Uffici Interessati

**1 (Esempio di Normalizzazione)**

### La Seconda Forma Normale

Riguarda le tabelle in cui la chiave primaria sia costituita da più di un campo. In questo caso Tutte le colonne corrispondenti agli attributi dipendono dall'intera chiave primaria.

#### **Esempio**

*Tabella singola*

Codice Città

Codice via

Città

Via

1. La chiave primaria è composta da Codice Città e Codice Via

2. Via dipende **sia** da Codice Città che da Codice Via
3. Città dipende **solo da Codice Città**: la colonna Città dipende quindi **solo da una parte** della Chiave primaria.
4. Per correggere l'errore è necessario dividere in due la Tabella (relazionandole): una **Tabella Città** ed una **Tabella Vie**: ognuno dei campi delle tabelle dipenderà esclusivamente dalla Chiave primaria della rispettiva tabella (Codice Città e Codice Via).

### Soluzione

Tabella Vie	Tabella Città
<u>Codice Città</u> <u>Codice via</u> Via	<u>Codice Città</u> Città

### Esempio grafico

Codice Città	Codice Via	Città	Via
01	01	Roma	Verdi
01	02	Roma	Bianchi
01	03	Roma	Rossi
02	01	Milano	Gialli

Codice Città	Codice Via	Via
01	01	Verdi
01	02	Bianchi
01	03	Rossi
02	01	Gialli

Codice Città	Città
01	Roma
02	Milano
03	Torino
04	Genova

**2 (Esempio di Normalizzazione)**

### La Terza Forma Normale

Stabilisce che non esistono dipendenze tra colonne di una tabella se non basate sulla chiave primaria.

#### Esempio

*Prima Tabella*

N° Protocollo

Mittente

Tipo

Urgenza

*Seconda Tabella*

Tipo

Descrizione

Nel caso della Prima Tabella Abbiamo che la Colonna (campo) *Urgenza* non dipende dalla campo chiave primaria , ovvero il numero di Protocollo ma bensì dal campo *Tipo* che è la chiave primaria della seconda tabella.

## Soluzione

Tabella Via	Tabella Città
N° Protocollo	Tipo
Mittente	Descrizione
Tipo	Urgenza

Anche in questo caso la dipendenza va ricondotta modificando le tabelle.

<table border="1"><thead><tr><th>N° Protocollo</th><th>Mittente</th><th>Tipo</th><th>Urgenza</th></tr></thead><tbody><tr><td>1</td><td>001</td><td>1</td><td>Si</td></tr><tr><td>2</td><td>001</td><td>2</td><td>No</td></tr><tr><td>3</td><td>003</td><td>1</td><td>Si</td></tr><tr><td>4</td><td>002</td><td>2</td><td>No</td></tr><tr><td>5</td><td>003</td><td>3</td><td>Si</td></tr></tbody></table>	N° Protocollo	Mittente	Tipo	Urgenza	1	001	1	Si	2	001	2	No	3	003	1	Si	4	002	2	No	5	003	3	Si	<table border="1"><thead><tr><th>Tipo</th><th>Descrizione</th></tr></thead><tbody><tr><td>1</td><td>Lettera</td></tr><tr><td>2</td><td>Memo</td></tr><tr><td>3</td><td>Telegramma</td></tr></tbody></table>	Tipo	Descrizione	1	Lettera	2	Memo	3	Telegramma	<table border="1"><thead><tr><th>Tipo</th><th>Descrizione</th><th>Urgenza</th></tr></thead><tbody><tr><td>1</td><td>Lettera</td><td>Si</td></tr><tr><td>2</td><td>Memo</td><td>No</td></tr><tr><td>3</td><td>Telegramma</td><td>Si</td></tr></tbody></table>	Tipo	Descrizione	Urgenza	1	Lettera	Si	2	Memo	No	3	Telegramma	Si
N° Protocollo	Mittente	Tipo	Urgenza																																											
1	001	1	Si																																											
2	001	2	No																																											
3	003	1	Si																																											
4	002	2	No																																											
5	003	3	Si																																											
Tipo	Descrizione																																													
1	Lettera																																													
2	Memo																																													
3	Telegramma																																													
Tipo	Descrizione	Urgenza																																												
1	Lettera	Si																																												
2	Memo	No																																												
3	Telegramma	Si																																												

**3 (Esempio di Normalizzazione)**

### Esempio grafico

Esistono molte altre forme normali, da utilizzare in un rigoroso processo di verifica del nostro progetto di database, ma non vengono in genere affrontate perché consentono un lavoro di "affinamento" del nostro progetto, che va oltre le più comuni e basilari operazioni di controllo.

## Relazioni e Cardinalità in Access

Quello che dal punto di vista pratico verrà rappresentato in Access 2003 con le relazioni tra dati e tabelle, va prima compreso dal punto di vista della Cardinalità.

Questa parola permette di esprimere il concetto di legame tra gli elementi di diversi insiemi di dati.

Le diverse entità del database possono avere legami tra loro espresse da relazioni dette *relation set*. Le relazioni tra le diverse entità vengono rappresentate in base alla loro **cardinalità**. Immaginiamo di avere due insiemi A e B:

- **cardinalità 1:1 (uno a uno)** - significa che ad un elemento di A può corrispondere uno ed uno solo elemento di B e viceversa  
Esempio: due entità di un database (tabelle) Nazioni e Capitali: ad una nazione può corrispondere solo una capitale e viceversa
- **cardinalità 1:N (uno a molti)** - significa che ad un elemento di A possono corrispondere più elementi di B ma non viceversa  
Esempio: due entità di un database (tabelle) Madri e Figlio: ad una Madre possono corrispondere più figli, ma un figlio può avere una sola madre
- **cardinalità N:N (molti a molti)** - significa che ad un elemento di A possono corrispondere più elementi di B e viceversa  
Esempio: due entità di un database (tabelle) Ordini e Articoli: Un ordine può contenere più articoli e viceversa lo stesso articolo può apparire in più ordini.

## Progettazione di un DatabaseE: un lungo cammino...

Prima di poter creare un database sul nostro computer utilizzando un software come Access 2003, un potente RDBMS, occorre analizzare e comprendere le fasi e le operazioni che portano a progettarlo, potremmo dire ancora prima a "pensarlo".

In genere questo processo da punto di vista logico attraversa una serie di fasi che permettono di passare dalla fase più astratta, l'idea del nostro database fino a quella più concreta relativa alla sua creazione.

Per progettare una base dati occorre stabilire il **modello** che determinerà le regole per la giusta rappresentazione dei dati attraverso uno **schema**, che sarà articolato in tre livelli:

### 1. Schema concettuale

E' la fase più astratta svincolata dal database vero è proprio che si andrà a costruire. I modelli impiegati in questo tipo di teorizzazione sono definiti modelli semantici. Il più noto è il modello **Entity-Relationship** (E-R Entità - Relazione). Questo modello prevede l'individuazione delle **entità** base concrete ed astratte che andranno a formare insiemi di entità (entity set). Ogni entità avrà i suoi **attributi**. Per comprendere meglio il concetto si può immaginare l'entità di un database rappresentata da una *tabella*, con i *campi* a formare gli *attributi*.

### 2. Schema logico

In questa fase di sviluppo il modello semantico entity-relationship viene espresso attraverso uno schema logico. Lo schema logico sarà vincolato dallo strumento DBMS, che verrà utilizzato per creare il database fisico, vero e proprio.

Lo schema logico si occupa di stabilire i modelli di relazione, di accesso e di interrogazione dei dati. Da questo punto di vista si possono distinguere più tipi di database, piatto, gerarchico e relazionali. L'ultima frontiera supera i database relazionali e parla di database a oggetti.

Database piatto: tabelle e record non sono collegati. E' impossibile creare strutture di dati relazionati

Database gerarchico: i dati sono collegati utilizzando puntatori complessi. Il sistema è difficilmente gestibile ed integrabile. Richiede l'opera di un programmatore esperto.

Database relazionale: i dati possono essere collegati tra loro con semplici relazioni impostate attraverso il linguaggio SQL.

La traduzione dello schema relazionale deve essere fatto usando un RDBMS il quale deve supportare il **linguaggio SQL** (Structured Query Language). Questo linguaggio è perfettamente adatto ad *implementare lo schema logico* di un database attraverso la creazione di tabelle, query, relazioni. Si può dire che l'SQL è lo strumento per realizzare in pratica (schema fisico) ciò che viene progettato come schema logico.

### 3. Schema fisico

Prevede la costituzione del database su di una specifica architettura dotata di uno specifico sistema operativo

Normalizzazione dei dati

Integrità referenziali

## DBMS e RDBMS

La gestione dei database ha visto notevoli evoluzioni nel corso degli ultimi 30/40 anni passando da tipologie di gestione dei dati molto complicate e riservate ad abili programmatori, a gestioni semplificate grazie a programmi come MSAccess in grado di agevolare il lavoro su database anche ad utenti poco pratici.

Il termine DBMS ovvero "Data Base Management System" si riferisce a quei programmi che permettono la gestione di database.

Con il più recente RDBMS ovvero "Relational Database Management System" abbiamo l'introduzione della gestione delle relazioni che intercorrono tra i dati (tabelle)  
Considerare oltre ai dati presenti nelle tabelle anche le relazioni che tra questi possono intercorrere, trasforma lo strumento da DBMS a RDBMS. MSAccess è un RDBMS a tutti gli effetti.

Ma quale è stata l'evoluzione dei database?

Tipologie di database

- Database piatto: tabelle e record non sono collegati. E' impossibile creare strutture di dati relazionati
- Database gerarchico: i dati sono collegati utilizzando puntatori complessi. Il sistema è difficilmente gestibile ed integrabile. Richiede l'opera di un programmatore esperto
- Database relazionale: i dati possono essere collegati tra loro con esempi relazioni impostate attraverso il linguaggio SQL
- Database ad oggetti: le nuove metodologie di gestione prevedono di gestire dati e metadati come oggetti su cui effettuare operazioni.

Ma quali sono i vantaggi di un RDBMS?

Indipendenza dei dati dalle Applicazioni

Nel caso in cui sia presente un DBMS le applicazioni sviluppate non dovranno occuparsi della struttura dei dati, non dovranno conoscerla e non dovranno occuparsi delle sue possibili modifiche. Questo compito è svolto dal DBMS. L'applicazione è svincolata dal dato.

Riservatezza dell'accesso ai Dati

Un sistema basato su un DBMS consente visioni logiche parziali dei dati. Alcune categorie di utenti possono avere visioni parziali degli archivi. Cosa impossibile nel caso in cui i dati risiedano sul filesystem. In questo caso una vista logica comporterebbe la modifica dell'intera struttura dell'archivio.

Gestione dell'integrità fisica dei Dati

Un DBMS garantisce la persistenza dei dati, anche in caso di accesso erroneo o caduta di tensione del sistema. Inoltre il sistema deve proteggere i dati da un possibile accesso contemporaneo, attraverso un meccanismo di duplicazione automatica e trasparente del dato (lock del campo)

Gestione dell'integrità logica dei Dati

Un DBMS consente di impostare dei vincoli tra i dati, che impediscano modifiche che non hanno senso nel contesto della logica rappresentata. Il caso tipico è rappresentato dall'integrità referenziale che previene cancellazioni o modifiche tra dati appartenenti a insiemi diversi (tabelle relazionate) e quindi sottostanti regole ben precise.

Sicurezza e ottimizzazione nell'uso dei Dati

Un grosso vantaggio nell'uso del DBMS è la possibilità di centralizzare utenti e risorse. L'amministratore di sistema può definire dei vincoli di accesso ai dati, dando ad utenti differenti, differenti permessi di lettura o scrittura dei dati. Inoltre il DBMS può ottimizzare l'accesso diretto alle periferiche dialogando con il sistema operativo che le gestisce.

## **Dati e Matadati**

Per poter lavorare più agevolmente e comprendere meglio il significato di alcuni termini è utile introdurre un concetto essenziali quello del dato e del matadato. Per schematizzare i concetti e comprenderne meglio i significati possiamo definire:



- I metadati rappresentano lo schema della base dati. Includono una serie di definizioni che descrivono la struttura dei dati, le restrizioni sui valori accettati (vincoli di integrità), le relazioni fra gli insiemi, potremmo definirli come la struttura fisica dell'archivio.
- I dati invece rappresentano le informazioni memorizzate negli archivi e hanno caratteristiche precise. Sono organizzati in insiemi omogenei tra loro (tabelle) che possono intrattenere relazioni con altri insiemi. Hanno alcune caratteristiche distintive, sono molti e rappresentano il volume dell'archivio e soprattutto sono permanenti, quindi una volta creati risiedono nell'archivio fino a che non vengono esplicitamente rimossi.

Per semplificare potremmo dire che "se i dati sono il contenuto dell'archivio i metadati ne sono il contenitore".

Vediamo un esempio grafico che chiarisce meglio il contenuto:

**Metadati:** nomi dei campi, tipologia, proprietà e vincoli

**Dati:** record inseriti

Nome	Tipo	Proprietà
Cognome	Testo	50 caratteri
Nome	Testo	50 caratteri
Città	Testo	30 caratteri
CAP	Testo	5 Caratteri
Telefono	Testo	15 caratteri
Data nascita	Data	

Cognome	Nome	Città	Cap	Telefono	Data nascita
Pavan	Paolo	Collegno	10093	011402333	25451
Rossi	Mario	Torino	10100	011402334	27009
Verdi	Enrcio	Rivoli	10098	011402335	24263

## Uso della Tastiera in Access

### Spostamento tra record

- TAB: verso il record a Dx
- TAB + maiusc: verso il record a Sx
- CTRL+Freccia dx
- CTRL+Freccia sx
- CTRL+Pg Up: si passa alla prima riga il alto
- CTRL+Pg Down: si passa all'ultima riga il basso

### Immissione di Dati

- CTRL +Z: elimina il dato immesso nel campo, se tenuto premuto elimina tutti i campi del record corrente.
- CTRL+' : copia il campo soprastante
- CTRL+ HOME: muove al primo record della tabella
- CTRL+ FINE: muove all'ultimo record della tabella

### I simboli delle sezioni

- *Triangolo nero*: appare quando si seleziona un record
- *Matitina*: Appare nel momento in cui si sta modificando un record
- *Stop*: indica che il record è bloccato
- *Asterisco*: indica il nuovo record

### Puntatori Foglio Dati

- *Freccia bianca*: tradizionale, usato per selezionare i comandi

- *Piccola Freccia Nera verso il basso*: compare sopra le intestazione di Riga e di Colonna, consente la sezione di colonne e o record o gruppi di record.
- *Doppia freccia*: consente di allargare o restringere righe o colonne
- *Croce bianca*: appare quando si selezionano campi, anche di record differenti all'interno del foglio di lavoro.

### **Puntatori Maschera**

- *Manina*: consente di muovere la casella di testo e la sua etichetta
- *Manina con indice*: consente di muovere gli elementi separati
- *Freccia diagonale*: consente di allargare o restringere un controllo.

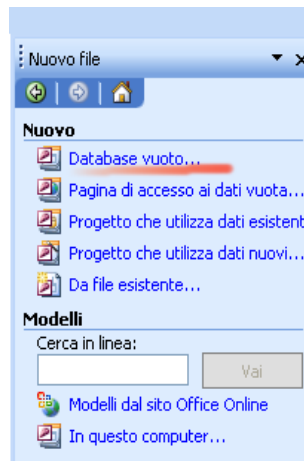
## **Creazione di un database**

Se abbiamo una precisa idea del tipo di dati che vogliamo rappresentare e archiviare possiamo procedere alla creazione del nostro database.

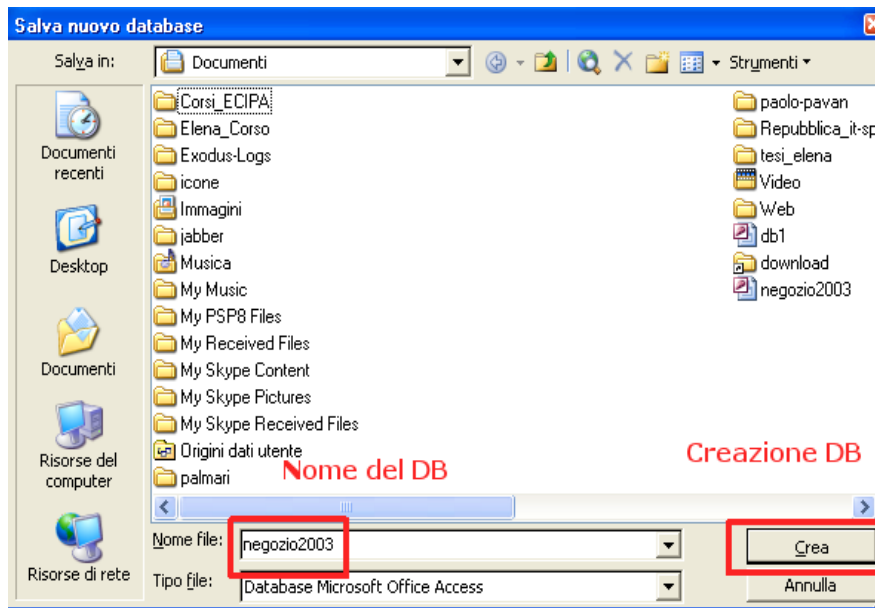
Una delle differenze rispetto agli altri programmi della suite di MSOffice è che nel loro caso si potevano modificare i dati agendo su di un documento non ancora salvato o creato. Nel caso di Access è bene diverso in quanto il programma richiede prima di tutto di creare il file, il database in cui verranno inseriti i nostri dati:

Una volta eseguito il programma ci si trova davanti ad un interfaccia a menu da cui possiamo scegliere il comando:

**Menu File >> Nuovo >> Database Vuoto**



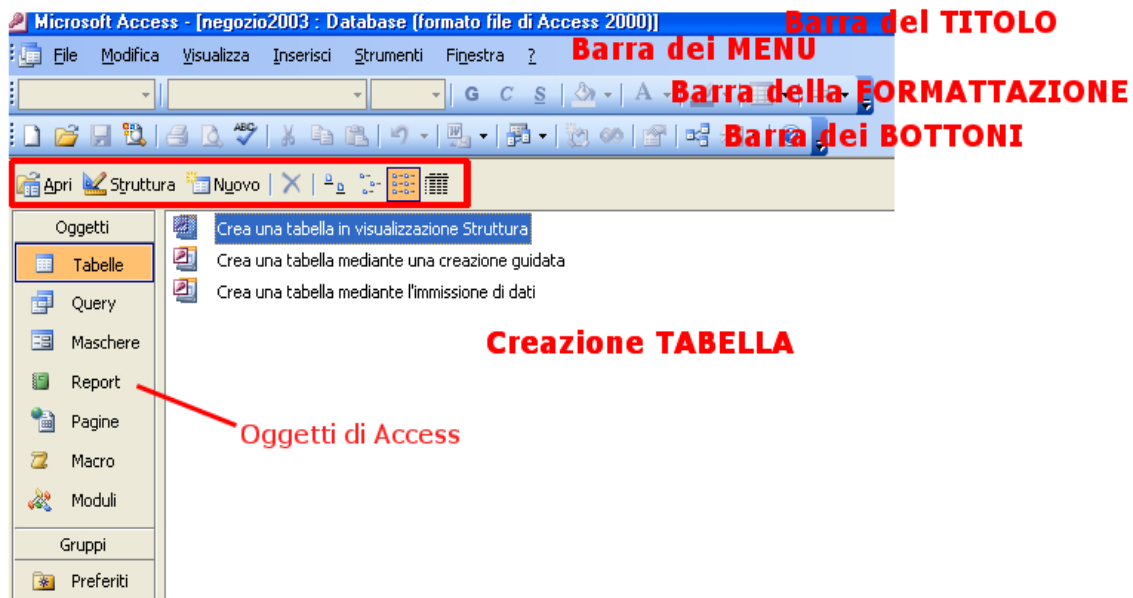
La voce "Database Vuoto" è la scelta da cui partire per poter creare il contenitore per i nostri effettivi dati.



## Interfaccia di Access

Una volta aperto il nostro database ci troviamo di fronte ad un'interfaccia menu abbastanza classica e tipica dei programmi di Office.

In realtà le differenze ci sono e sono molte. Ad esempio dobbiamo ricordarci che vista la complessità del programma e la quantità di dati presenti, le voci di menu si modificano e si adattano all'oggetto o all'operazione che stiamo svolgendo. Non deve quindi spaventare il fatto di non trovare una voce nel menu, questo deriva sicuramente da un'errata selezione del contesto in cui dobbiamo agire.



Gli elementi presenti nell'interfaccia sono:

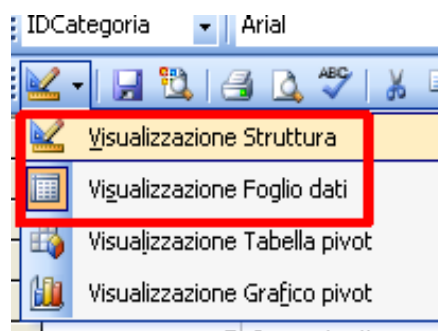
- **Tabelle:** rappresentano i contenitori per i diversi record. Sono essenziali per la costruzione del database
- **Query:** rappresenta un'interrogazione della base dati. Viene effettuata selezionando il campo o i campi desiderati da una o più tabelle. Se esiste un legame tra le tabelle, la query sarà di tipo relazionale, permetterà quindi di estrarre dati correlati tra più tabelle. La risposta alla query viene chiamata set di risultati, compatibili ai criteri imposti dalla query

- **Maschere:** sono moduli visualizzati sullo schermo che facilitano l'immissione e la lettura dei dati. Rispetto alle tabelle hanno una serie di funzionalità aggiuntive, come l'uso di strumenti grafici per potenziare e facilitare l'analisi dei dati inseriti
- **Report:** in genere rappresentano copie su carta dei set di risultati. Se la trattazione dei dati deve avvenire a video si preferisce utilizzare la maschera, ma se i dati devono essere stampati allora è meglio creare un report adatto alle nostre esigenze
- **Pagine:** le pagine di accesso ai dati sono pagine esportabili su web che permettono di costruire moduli web, consultabili e aggiornabili direttamente con il Browser Internet Explorer.
- **Macro:** sono strumenti che automatizzano operazioni manuali complesse, in genere ripetitive o molto lunghe. Spesso creare una macro può significare un costante risparmio di tempo, nonché la certezza di non commettere errori
- **Moduli:** sono programmi scritti utilizzando il linguaggio access basic. Permettono di scrivere interfacce per il database stesso integrandolo e potenziandolo.

Ogni elemento presenta delle visualizzazioni diverse che consentono tipi di operazioni diverse. Un elenco completo presenta queste visualizzazioni:

ELEMENTI	VISUALIZZAZIONI	VISUALIZZAZIONI	VISUALIZZAZIONI
<b>Tabelle</b>	Struttura	Fogli Dati	
<b>Maschere</b>	Maschera	Struttura	Fogli Dati
<b>Query</b>	Struttura	SQL	Fogli Dati
<b>Report</b>	Struttura	Anteprima Stampa	
<b>Macro</b>	Struttura		

Ogni oggetto selezionato, come ad esempio una tabella, presenta nella sua barra in alto un bottone che permette di cambiare tipo di visualizzazione



## Analisi delle Tabelle database

Dal punto di vista pratico un database è costituito da un insieme di singoli dati che per ragioni di correttezza, come già detto, non devono essere duplicati. Gli elementi essenziali di un database sono a partire dal livello inferiore:

- **campi:** sono le voci che costituiscono un record
- **record:** rappresenta la registrazione di un singolo dato, descritto da più campi
- **tabelle:** sono costituite dall'insieme dei record
- **database:** è costituito dalla singola tabella o da più tabelle eventualmente tra loro relazionate

Com_Cod	Com_Descrizione	Com_Costo	Lab_Cod
0010	Montante laterale 100 cm	50000	0010
0020	Montante laterale 200 cm	100000	0010
0030	Ripiano 60 cm	50000	0050
0040	Piano tavolo tondo	120000	0050
0050	Piano tavolo quadrato	120000	0050
0060	Gamba tavolo 60 cm	20000	0020
0070	<del>Gamba tavolo 40 cm</del>	<del>Record</del>	
0080	Anta 100 cm	80000	0040
0090	Anta 200 cm	120000	0040
0100	Cassettiera da incasso	200000	0030
0110	Busta 10 tasselli	5000	
0120	Busta 10 viti	10000	
0130	Pomello anta/cassetti	5000	0060
0140	Bastone appendiabiti	15000	0060
0150	Pannello posteriore 100 cm x 60	30000	0050

Vediamo la corretta nomenclatura per tutti gli elementi delle tabelle;

**Tabella, Entità, Relazione**

**Record, Riga, Istanza, Tupla**

**Campo, Attributo, Colonna**

**Relazione, Associazione, Legame**

## Creazione del database

Cominciamo con il creare le due tabelle essenziali per la catalogazione dei dati nel nostro database.

Selezionando la voce tabella e poi "Crea una tabella in Autocomposizione guidata"

A questo punto occorre inserire nomi e caratteristiche dei campi, operazione che ci consente di predisporre al meglio il nostro contenitore di dati.

Nome campo	Tipo dati	
IDProdotto	Contatore	Numero assegnato automaticamente a ogni nuovo prodotto.
NomeProdotto	Testo	
IDCategoria	Numerico	Uguale voce nella tabella Categorie.
QuantitàPerUnità	Testo	Per esempio: confezione da 24 pezzi, bottiglia da 1 litro.
PrezzoUnitario	Valuta	
Scorte	Numerico	<b>Struttura TABELLA</b>
QuantitàOrdinata	Numerico	
LivelloDiRiordino	Numerico	Quantitativo minimo da tenere in magazzino.
Sospeso	Sì/No	Sì, indica che il prodotto non è più disponibile.

I campi da inserire per la tabella Categorie con il loro tipi di campi

1. IDCategoria: Contatore
2. NomeCategoria: Testo
3. Descrizione: Memo
4. Immagine: Bitmap

I campi da inserire per la tabella Prodotti con il loro tipi di campi

1. IDProdotto: Contatore
2. IDCategoria: Numerico
3. NomeProdotto: Testo
4. QuantitàPerUnità: Testo
5. PrezzoUnitario: Valuta
6. Scorte: Numerico
7. QuantitàOrdinata: Numerico
8. LivelloDiRiordino: Numerico
5. Sospeso: Sì/No

→ il campo IDCategoria viene inserito come relazione

I campi da inserire per la tabella Ordini con il loro tipi di campi

1. IDOrdine: Contatore
2. IDCliente: Numerico
3. DataOrdine: Data
4. Destinatario: Testo
5. PaeseDestinatario: Testo

→ il campo IDCliente viene inserito come relazione

I campi da inserire per la tabella Clienti con il loro tipi di campi

1. IDCliente: Contatore
2. NomeProdotto: Testo
3. QuantitàPerUnità: Testo
4. PrezzoUnitario: Valuta
5. Scorte: Numerico
6. QuantitàOrdinata: Numerico
7. LivelloDiRiordino: Numerico

I campi da inserire per la tabella DettagliOrdini con il loro tipi di campi

1. IDOrdine: Contatore
2. IDProdotto: Contatore
3. Quantità: Numerico
4. Sconto: Numerico

→ Il campo IDOrdine e IDProdotto vengono inseriti come relazione.

In pratica questa tabella viene create per risolvere il problema della relazione molti a molti che intercorre tra le tabelle Ordini e Prodotti. In pratica la relazione N a N si spezza in due relazioni 1 a N, che entrambe le tabelle Ordini e prodotti vengono ad avere con la tabella DettagliOrdini

## Descrizione dei tipi di Dati

Tipo di Dati	
<b>Testo</b>	Supporta caratteri alfa-numeric. E' lungo da 1 a 255 caratteri. Il default è 50
<b>Memo</b>	Supporta caratteri alfa-numeric. Può ospitare fino a 64.000 caratteri non possono essere indicizzati
<b>Numerico</b>	Utilizzato per immettere solo numeri o per fare calcoli sui campi. Può avere dimensioni e quindi precisioni differenti.
<b>Data/Ora</b>	Utilizzato per inserire la data in formati differenti
<b>Valuta</b>	Utilizzato per inserire valori che riguardano importi di denaro
<b>Contatore</b>	E' il campo utilizzato per numerare automaticamente i record immessi. <b>Spesso coincide con il campo Chiave Primaria</b>
<b>Si/No</b>	Utilizzato per inserire dei valori tipo vero/Falso oppure Maschi/Femmina, Pagato/Insoluto.
<b>Oggetto OLE</b>	Consente di Inserire un oggetto OLE (Immagine o altro) da associare al record.
<b>Collegamento Ipertestuale</b>	Tipo specifico per inserire un collegamento ad un sito internet, ad esempio ad un indirizzo email che apre automaticamente il vostro programma di posta elettronica, se installato.
<b>Autocomposizione Ricerca</b>	Consente di inserire un campo che mostri dei valori da un elenco riepilogativo, ricavato da un'altra tabella oppure creato manualmente.

Ogni campo può avere una serie di caratteristiche tra cui in vincoli di inserimento, come il tipo di dato la dimensione, il valore fino ad arrivare a complesse "maschere di input".

**Proprietà dei CAMPI**

Generale	Ricerca
Dimensione campo	Intero lungo
Nuovi valori	Incremento
Formato	
Etichetta	ID prodotto
Indicizzato	Si (Duplicati non ammessi)
Smart tag	

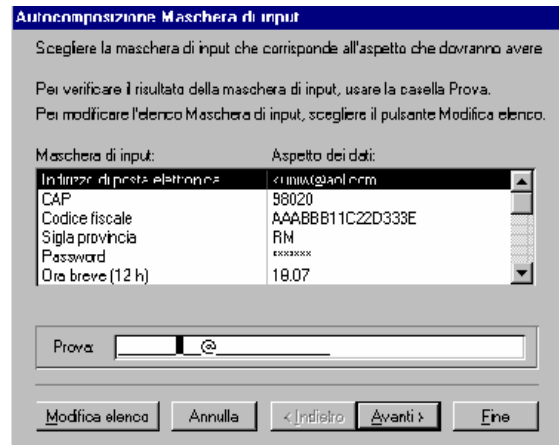
Ogni tipo di Dati presenta le sue **specifiche caratteristiche** che sono comunque simili per tutti. Vediamo in rapida sequenza una descrizione delle diverse proprietà dei campi della nostra tabella.

Generale	Ricerca
Dimensione campo	15
Formato	
Maschera di input	
Etichetta	Nome categoria
Valore predefinito	
Valido se	
Messaggio errore	
Richiesto	Si
Consenti lunghezza zero	No
Indicizzato	Si (Duplicati non ammessi)

- **Dimensione Campo:** è la lunghezza in cifre del campo
- **Formato:** è possibile vincolare il formato d'inserimento con simboli specifici  
Ad es: inserire in questo campo > vuol dire che il testo andrà MAIUSCOLO  
inserire in questo campo < vuol dire che il testo andrà minuscolo

usare @ come segnaposto per le cifre  
usare & come segnaposto per cifre e numeri  
usare - per dividere le parole

- **Maschera di Input:** Consentono di vincolare l'inserimento dei dati offrendo un modello (maschera) da rispettare obbligatoriamente per inserire i dati. Si creano con una pratica autocomposizione:



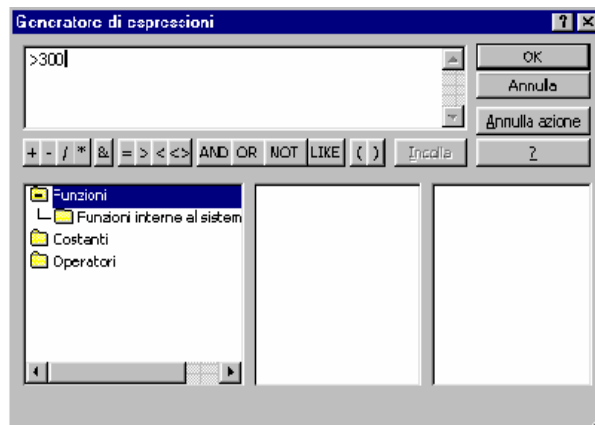
Basta scegliere il modello adatto ed effettuare anche una prova, all'interno dell'apposita casella.

L'autocomposizione procede consentendo anche all'utente di modificare alcuni aspetti della maschera stessa a seconda delle proprie esigenze.



- **Etichetta:** consente di inserire un'etichetta che sostituisce il nome del campo nella Visualizzazione Foglio Dati senza costringerci a cambiare il nome di un campo. In genere si usa per tenere il nome vero del campo compatto, mentre con l'etichetta gli si può assegnare un nome esteso.
- **Valore Predefinito:** Consente di fare comparire nel campo un valore di default, immesso automaticamente, che poi può ovviamente essere modificato.
- **Valido Se:** consente di impostare una condizione di inserimento dati. Si possono generare attraverso il **Generatore di Espressioni**, vere e proprie espressioni che possono vincolare l'inserimento di particolari dati. Nell'esempio il campo accetterà solo valori superiori a 300 e non consentirà di andare oltre prima di aver corretto un eventuale inserimento sbagliato.





- **Messaggio di Errore:** Consente di personalizzare il messaggio di errore che viene visualizzato quando non vengono rispettate le condizioni di validità.
- **Richiesto:** questo campo Si/No viene utilizzato per obbligare a compilare il campo in ogni caso.
- **Consenti Lunghezza Zero:** permette di inserire un valore a lunghezza zero per il campo. Si utilizza nel caso in cui il campo possa ospitare un valore nullo.
- **Indicizzato:** se viene assegnato ci consentirà di effettuare le ricerche in maniera più rapida.

Nel caso in cui il campo sia una **Chiave Primaria**, sarà assegnato automaticamente la voce **Duplicati non Ammessi**, perché il campo chiave primaria che identifica **univocamente** il Record **non può essere mai ripetuto**.

**Attenzione!.** A seconda del tipo di Dati scelto alcuni di questi campi potranno apparire o meno

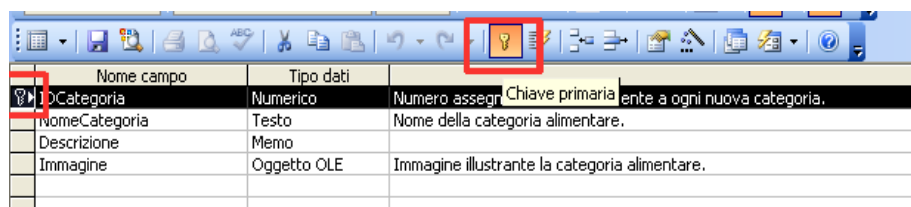
## Definizione della Chiave Primaria

Un RDBMS necessita sempre di un modo per identificare in maniera univoca ciascun record che viene inserito nella tabella. Questo elemento è conosciuto come Chiave Primaria

La chiave primaria viene identificata con un campo (o con un insieme di campi) che rendono unico il record inserito. In genere il campo scelto come Chiave Primaria della tabella non può mai essere duplicato, quindi non si può ripetere (è unico). I campi che possono essere utilizzati come chiavi primarie vengono chiamati Chiavi Candidate.

Un campo di chiave primaria non può mai contenere dati duplicati e non può essere lasciato vuoto.

Per definire le chiavi primarie Accedere alle singole tabelle e selezionare la voce IDCategoria, IDProdotto, IDClienti, IDOrdine e cliccare sul bottone indicante la "chiavetta" sulla barra dei bottoni:



Questa operazione va fatta partendo da tutte le tabelle in "visualizzazione struttura" e cliccando sulla chiavetta. In questo caso i campi selezionati non potranno essere duplicati, per questo sono campi di tipo Contatore e definiranno in modo univoco il singolo record inserito nella tabella.

## Creazione delle Relazioni

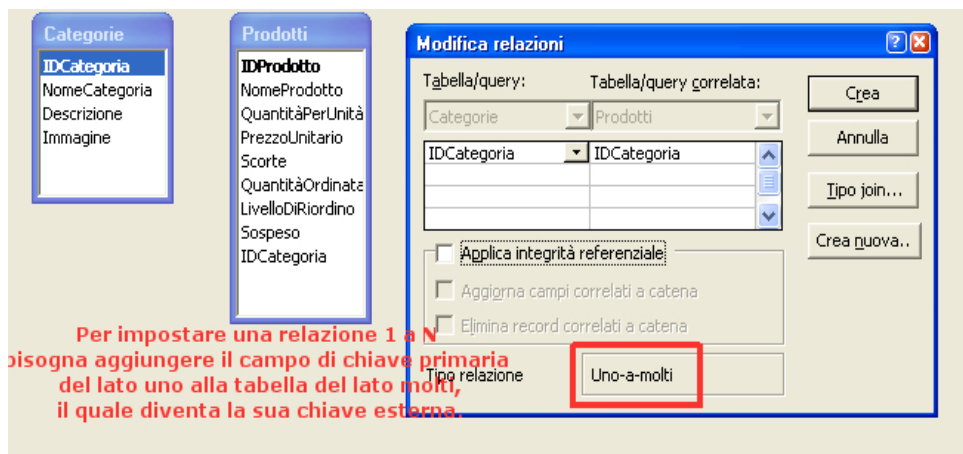
Per creare le relazioni tra le nostre tabelle ci sono diversi metodi. Il sistema più corretto ed organico prevede di cliccare su

**Menu Strumenti >> Relazioni**

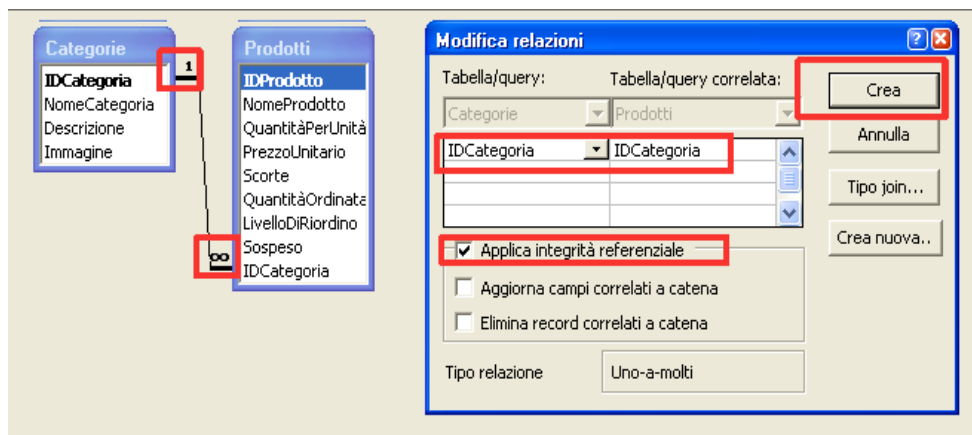
A questo punto cliccando sul Menu

**Menu Relazioni >> Mostra Tabelle**

Aggiungiamo le nostre tabelle

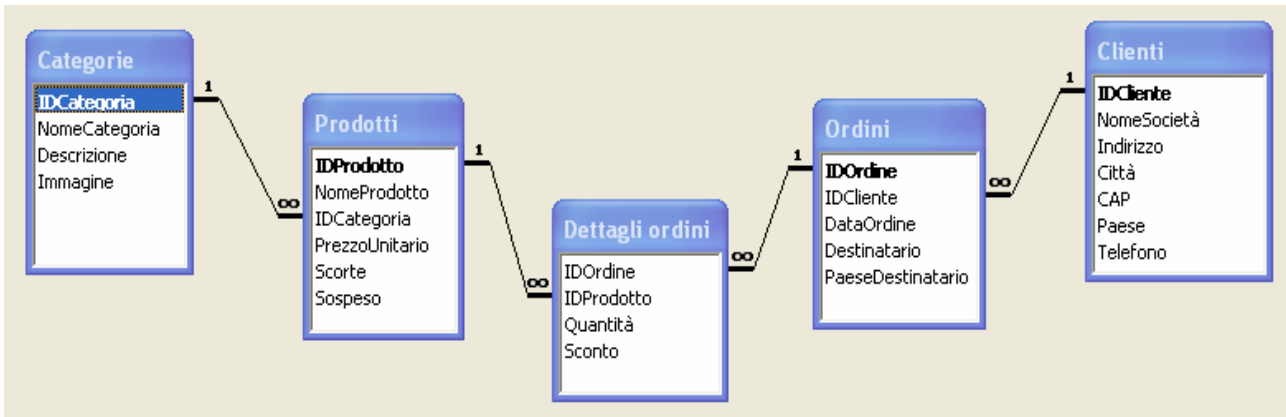


Trascinando il campo Chiave primaria del lato molti (IDCategoria), sulla tabella del lato uno (Prodotti) sull'analogo campo IDCategoria impostiamo la relazione:



Questa Operazione va fatta per la relazione tra la tabella Clienti e la tabella Ordini. Un discorso a parte merita invece la relazione tra Ordini e Prodotti che essendo di tipo N a N (molti a molti) deve essere risolta, creando una nuova tabella che consenta di avere due nuove relazioni del tipo uno a molti.

La situazione finale del nostro database con le sue relazioni deve essere uguale all'immagine sottostante.



Il legame tra due tabelle viene definito Join. La chiave primaria della tabella del lato uno che viene aggiunta al lato molti viene definita in quest'ultima tabella "chiave esterna".

### Uso dell'Autocomposizione ricerca per la definizione delle relazioni

Un altro metodo per semplificare la definizione delle relazioni è quello dell'uso del campo "Autocomposizione ricerca".

→ da completare

### Definizioni degli Indici

Un indice è una struttura ausiliaria che serve per fare le ricerche in modo più rapido, cosicché l'utente debba aspettare meno le risposte. Essendo una struttura, che deve essere aggiunta al database e memorizzata sul disco (struttura ausiliaria), ha lo svantaggio di incrementare la memoria totale dell'archivio e di appesantire l'inserimento dei dati.

Utilizzando gli indici si crea una struttura "a parte", rispetto alla tabella, che tiene gli elementi in ordine.

	COGNOME	NOME	CITTA'
1	Verdi	Giagomo	Treviso
2	Rossi	Felice	Treviso
3	Verdi	Mario	Treviso
4	Gialli	Umberto	Milano
5	Rosa		
6	Verdi		
7	Bianchi		
8	Bianchi		
9	Rosa		
10	Gialli		

	Indice su Cognome	
7	Bianchi	7, 8, 13, 18, 24, 26, 34
4	Gialli	4, 10, 20, 27, 28, 31, 33, 36
5	Rosa	5, 9, 12, 17, 19, 21, 35
2	Rossi	2, 11, 15, 16, 22, 25, 30
1	Verdi	1, 3, 6, 14, 23, 29, 32

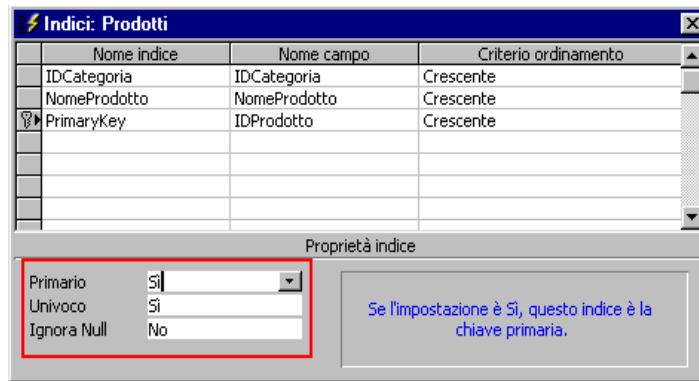
La ricerca sugli indici, essendo ordinati, è molto più rapida rispetto a quella sulla tabella: non si devono passare tutte le righe, **ma si cerca direttamente il valore nella posizione in cui dovrebbe trovarsi.**

Ad esempio si pensi all'elenco telefonico. La ricerca di un numero sull'elenco del telefono è molto simile a quella su un indice. La ricerca sulla tabella, invece, è simile a quella che si dovrebbe fare se l'elenco telefonico non fosse ordinato, ma i numeri fossero sistemati casualmente. Il paragone vale anche per il tempo impiegato a trovare i valori: nel primo caso si cerca direttamente in una posizione dell'elenco, nel secondo caso si deve partire dal primo numero e leggere tutti i seguenti fino a quando non si trova quello cercato.

**Posizione o coordinata del dato memorizzato sul disco del computer**

La ricerca sugli indici, essendo ordinati, è molto più rapida rispetto a quella sulla tabella: non si devono passare tutte le righe, ma si cerca direttamente il valore nella posizione in cui dovrebbe trovarsi.

Ad esempio si pensi all'elenco telefonico. La ricerca di un numero sull'elenco del telefono è molto simile a quella su un indice. La ricerca sulla tabella, invece, è simile a quella che si dovrebbe fare se l'elenco telefonico non fosse ordinato, ma i numeri fossero sistemati casualmente. Il paragone vale anche per il tempo impiegato a trovare i valori: nel primo caso si cerca direttamente in una posizione dell'elenco, nel secondo caso si deve partire dal primo numero e leggere tutti i seguenti fino a quando non si trova quello cercato.



- Ogni volta che viene assegnato "Si" al valore Indicizzato viene aggiunto un indice.
- La chiave primaria è indice di diritto, l'indice si dice Primario (Si)
- Se un indice ha valore Univoco questo non può essere duplicato
- Se "Ignora Null" è impostato a Si, l'indice non includerà i campi con valore nullo

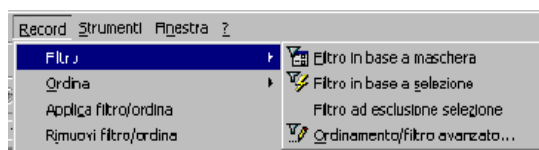
## Filtri e Ordinamento dei dati

### Filtri

Barra degli Strumenti

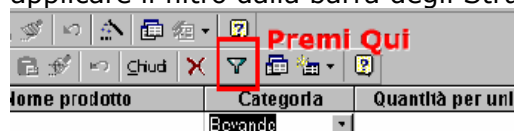


Nella Visualizzazione Foglio Dati o Visualizzazione Maschera, scegliere Strumenti /Filtro



#### 1. Filtro in Base a Maschera

Consente di inserire un elenco a discesa per ogni campo, effettuare un selezione e poi applicare il filtro dalla barra degli Strumenti.



#### 2. Filtro in base Selezione

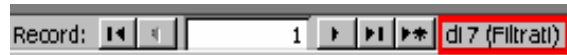
Applica il filtro isolando il record su cui si è posizionato il cursore. Se questo filtro si applica ad un campo ripetuto in altri record verranno evidenziati tutti record che lo contengono.

#### 3. Filtro ad esclusione di Selezione

Evidenzia tutti i record esclusi quelli che corrispondono (vengono filtrati) dal campo selezionato.

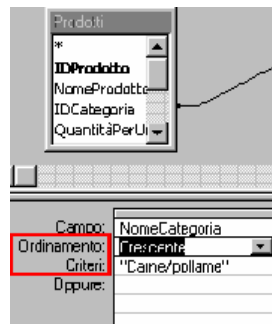
**Attenzione!** se il bottone del filtro è premuto sulla barra ripremelo per rimuoverlo e tornare alla totalità dei record.

Per capire se un filtro è applicato oltre al tasto premuto sulla barra, si avrà che il contatore in basso indicherà meno record rispetto al totale con la voce filtrati tra parentesi.



### Ordinamento Filtro Avanzato

Consente di applicare un filtro avanzato, applicando lo stesso concetto applicato per le query. Solo che invece di effettuare una query si applica un filtro. E' possibile allo stesso tempo ordinare con facilità i dati.



### Ordinamento Dati

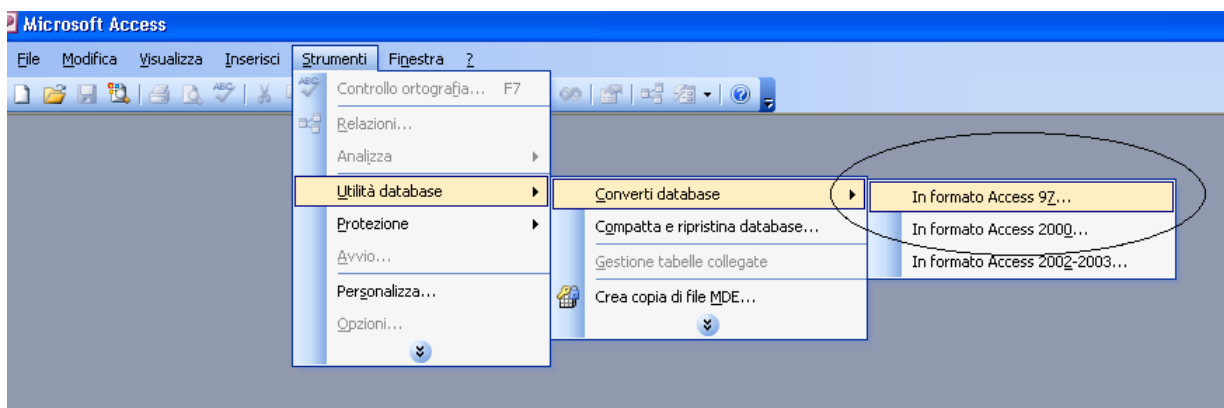
Utilizzando i bottoni presenti sulla barra degli strumenti si possono ordinare con facilità i dati per singoli campi.

Per fare ordinamenti su più campi (chiavi) contemporaneamente scegliere Ordinamento/filtro Avanzato.

Con la stessa Logica i filtri si possono applicare alle maschere cosa che è comunque possibile disabilitare.

## Formati di Access: conversioni dei database

Access supporta formati prodotti con versioni precedenti, richiedendo però la conversione al formato più recente oppure l'esportazione al formato più obsoleto.



Se si lavora con versioni recenti di Access occorre ricordarsi di convertire il DB per la versione con cui lo si dovrà utilizzare.

Questa operazione va fatta a DB chiuso e va scelta la versione verso cui effettuare la conversione (back port)

Attualmente sono tre:

- Formato Access 97
- Formato Access 2000
- Formato Access 2002-2003