Corso Database e Access

Dott. Paolo PAVAN

Netlink S.A.S
Via Alpignano 27
10093 collegno (TO)
pavan@netlink.it - http://www.netlink.it

Anno 2000-2002
Aggiornamento Ottobre 2003

Database
Concetto Semplice

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

EXCEL

ACCESS

Una base di dati, database o un archivio viene spesso erroneamente rappresentato come una semplice tabella.

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.

Database
Il giusto concetto

Cognome	Nome	Città
Rossi	Mario	01
Verdi	Giuseppe	02
Bianchi	Luigi	01
Rossi	Anna	02
Mazzini	Giuseppe	01
Mazzola	Valentino	03

Codice	Città	CAP
01	Torino	10100
02	Roma	12020
03	Mialno	12100

Database: 2 tabelle relazionate

Metadati: i contenitori e le loro strutture, proprietà e relazioni

Dati: il contenuto, organizzato per righe in insiemi omogenei, numerosi e permanenti (fino a che non vengono esplicitamente rimossi)

Requisiti

- **Access 97/2000/XP** e un RDBMS semplice da usare ma occorrono delle conoscenze base sull'utilizzo del sistema operativo MS Windows:
  - Aprire e chiudere un programma, applicazione o finestra in windows
  - Aprire e salvare un file
  - Passare da una finestra all'altra (multitasking)
  - Ingrandire, ridimensionare o portare a tutto schermo le finestre
  - Ridurre a icona o chiudere una finestra
  - Utilizzare il mouse, il puntatore e i menu
  - Utilizzare la tastiera
  - Utilizzare Esplora risorse e le Risorse del Computer: copiare, spostare rinominare un file e creare cartelle e sottocartelle.
  - Usare TAGLIA,COPIA e INCOLLA ed effettuare trascinamenti
  - Usare il tasto Destro del Mouse
- Occorre però conoscere **un po' di teoria** dei database che in genere i manuali non spiegano.

TEORIA delle Basi Dati
Terminologia

- **DATABASE**

  Indica in genere un **insieme di dati rivolti alla rappresentazione di un preciso sistema informativo** di vario tipo, aziendale, scientifico, amministrativo o altro.

- **Elementi essenziali**
  - La struttura dei dati
  - Le loro relazioni esistenti tra i dati
- **Rappresentazione schematica**
  - Schema CONCETTUALE
  - Schema LOGICO
  - Schema FISICO

Requisiti di un Database

- **REQUISITI (coerenza database)**
  - **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)

Progettazione Database

- **Schema concettuale**
  - Fase astratta
  - Modelli e teorizzazioni
  - Entity-relationship (E-R entità - relazione)

- **Modello entity-relationship**
  - 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

Entità base -->TABELLE

Attributi delle entità base --> CAMPI

Progettazione Database

- **Legami tra entità**
  - **Le entità possono avere dei legami (relazioni) rappresentate in base alla loro CARDINALITA'**
- **CARDINALITA' (relazioni)**
  - **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 Database

- **MODELLO SEMANTICO <u>entity-relationship</u> (schema concettuale) viene espresso attraverso uno schema logico**

- **Stabilisce i modelli di relazione, di accesso e di interrogazione dei dati**

Progettazione Database

- **STRUMENTO DBMS (data base management system)**
  - Lo schema logico sarà vincolato dallo strumento DBMS, che verrà utilizzato per creare il database fisico, vero e proprio
- **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 esempli relazioni impostate attraverso il linguaggio SQL

Progettazione Database

- **RDBMS: <u>Relational Database Management System</u>**
  - L'evoluzione della gestione dei database implica la 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:

    <u>DBMS</u> → **<u>RDBMS</u>**
  - Per gestire basi dati i cui elementi sono tra loro relazionati è necessario un software che utilizzi un linguaggio basato sull'algebra relazionale (linguaggio SQL)
  - Access, Oracle, Postgres, Ingres e Informix sono RDBMS relazionali.

Vantaggi Dell'uso 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 possibile 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.

Progettazione Database

- **SQL (structured query language)**
  - Consente di 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.

Istruzioni in SQL

- L'SQL è un linguaggio che consente di fornire *istruzioni* o comandi con i quali si formano enunciati. Alcuni comandi possiedono *clausole e predicati*.
- Questi comandi vengono suddivisi in 3 categorie:
  - ***DDL*** *(data definition language):* in questa categoria rientrano i comandi che servono per creare e modificare le tabelle e quindi per definire i dati. Ad esempio i comandi INSERT o UPDATE.
  - ***DML*** *(data manipulation language):* in questa invece ricadono i comandi con cui si manipolano i dati per ricavarne le informazioni necessarie. Ad esempio il comando SELECT.
  - ***DCL*** *(data control language):* in questa invece ricadono i comandi con cui stabiliscono priorità e accessi alle tabelle. Ad esempio i comandi GRANT o REVOKE.

Dialetti SQL

- L'American National Standard Institute (**ANSI**) presidia e cura le specifiche del linguaggio SQL a cui tutti gli RDBMS devono adattarsi se vogliono definirsi tali. Tre specifiche SQL:
  - **SQL1** emessa nel 1986
  - **SQL2** emessa nel 1992
  - **SQL3** emessa nel 1999: questo standard è stato fatto proprio anche dall'ISO (international standard organisation). Ha introdotto alcune novità come i cicli o le transazioni (ovvero la possibilità di eseguire più comandi in successione come se fossero un unico comando).
    - http://www.pluto.linux.it/ildp/howto/postgresql-howto-29.html

Strutturazione Database

- **Schema fisico**
  - Prevede la costituzione del database su di una specifica architettura dotata di uno specifico sistema operativo
    - **Normalizzazione dei dati**
    - **Integrità referenziali**

- **SOFTWARE per DB RELAZIONALI**
  - Traduce sul computer lo schema concettuale attraverso l'applicazione delle regole imposte dallo schema logico:
    - **ACCESS 95/97/2000/XP: applicativo per database su piattaforma Intel, con sistema operativo windows 95/98/2000/XP**
    - **Altri software concorrenti: Filemaker, Lotus Approach**

Riepilogo Schemi

Schema Concettuale

Rappresenta la mia idea, il mio progetto di gestione di un ipotetico insieme di dati

Schema Logico

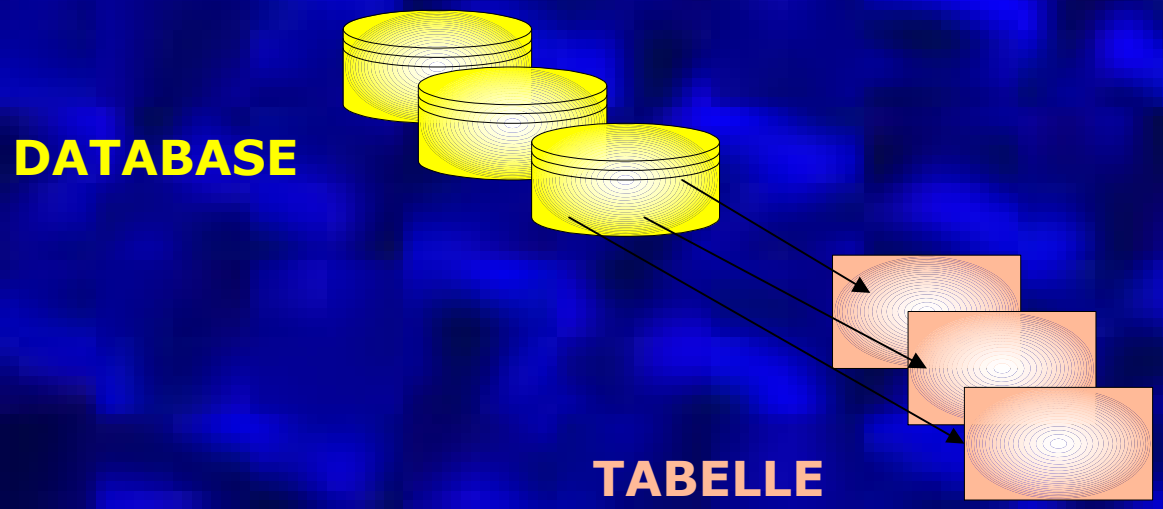
Ci permette di adottare una serie di strumenti logici per realizzare il nostro progetto.
Rappresenta la tecnica di organizzazione e accesso ai dati propria di specifiche categorie di DBMS.
Sarà vincolato alla tipologia del Linguaggio di programmazione (SQL) e di Database (Relazionale - RDBMS)

Schema Fisico

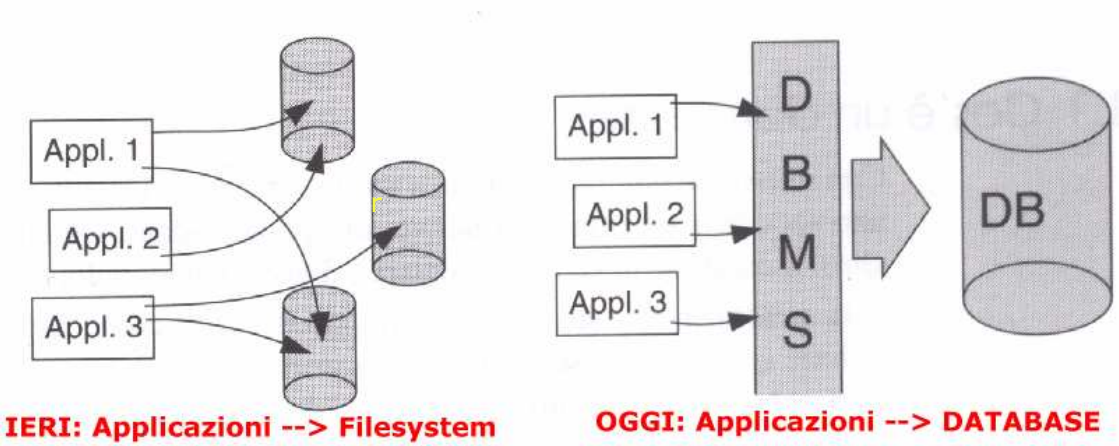
Coincide con la realizzazione pratica del mio progetto, attraverso uno strumento software ben preciso, che implementa un linguaggio di programmazione di un certo tipo, il tutto vincolato ad una piattaforma hardware/software ben definita.
Viene stabilito come le strutture definite a livello logico si organizzino in archivi sul filesystem.

Rappresentazione Grafica di un DATABASE

RDBMS



Accesso ai Dati



Accesso ai Dati: IERI

Negli anni sessanta, primo sviluppo dell'informatica l'accesso ai dati era orientato al Filesystem direttamente: lento, difficilmente organizzabile, richiedeva anche per operazioni semplici sforzi notevoli di programmazione.
In questo caso sono le applicazioni che accedono direttamente ai dati a dover conoscere la struttura dei dati e a rispettare i requisiti di ridondanza minima.

Accesso ai Dati: OGGI

Oggi invece le Applicazioni (tutte quelle presenti sul sistema, progettate per scopi diversi) accedono ad un unica base dati, che consente un facile utilizzo dei dati, anche in contemporanea e per le operazioni più complesse.
In questo modo **i dati sono centralizzati** e le applicazioni sono alleggerite, visto che sono svincolate, delle funzionalità che sono svolte da database, secondo funzioni standard. Quindi l'accesso avviene attraverso un'interfaccia rappresentata dal DBMS, a cui sono demandate gran parte delle operazioni che in precedenza erano a carico delle Applicazioni.

Utenti dei Database

Utenti Finali

Utilizzano le applicazioni che a loro volta dialogano con il DB.
Vedono i dati attraverso il filtro di tali applicazioni.
Utilizzano linguaggi di mediazione come l'SQL (estensione del DML) per interrogare i dati.

Sviluppatori

Realizzano i programmi destinati agli utenti finali, utilizzando il DML per la parte di programma relativa all'interazione con i dati.

Amministratori DBA

Sono i soggetti che avendo partecipato alla progettazione del database lo mantengono in esercizio, occupandosi degli aspetti relativi all'accesso ed alla cura dei dati (backup).
Utilizzano il DDL per progettare la struttura dei dati ed il DCL per le operazioni di controllo dei dati.

Dati e Metadati

- 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 organizzate in insiemi omogenei tra loro (tabelle) che possono intrattenere relazioni con altri insiemi.
  - Sono molti e rappresentano il volume dell'archivio
  - Sono permanenti: una volta creati risiedono nell'archivio fino a che non vengono esplicitamente rimossi.

Potremmo dire che se i dati sono il contenuto dell'archivio i metadati ne sono il contenitore.

Schema Metadati-Dati

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	

Metadati: nomi dei campi, tipologia, proprietà e vincoli

Dati: record inseriti

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

La Progettazione pratica di un Database: Come bisogna operare?

- La progettazione si occupa della costruzione dello schema, cioè dei metadati. La progettazione parte dall'idea o dalla necessità di costruire un archivio, e arriva fino alla sua creazione pratica.
  La creazione di un database può essere suddivisa in tre fasi:
  - **ANALISI DEI REQUISITI**: si stabilisce cosa si vuole rappresentare esattamente, lo scopo del mio progetto, lo si stabilisce con il cliente, dopo una accurata indagine;
  - **PROGETTO DEL SISTEMA**: si progetta un sistema basandosi sulle informazioni ricavate dall'analisi dei requisiti;
  - **REALIZZAZIONE DEL SISTEMA**: viene sviluppato direttamente con Access.

Analisi, Progettazione e Realizzazione con Access

- Prima di costruire un database dobbiamo **progettarlo**
- Occorre valutare con attenzione la **STRUTTURA** necessaria prima di costruirla praticamente: è bene tracciarne dettagliatamente i particolari su di un foglio di carta prima di cominciare a costruire un nuovo mdb.
- Una volta assicurati che **tutte le funzioni richieste e tutti i dati necessari** sono memorizzabili nella struttura creata, allora possiamo cominciare con lo sviluppo vero e proprio.
- **Modifiche sulle strutture** tabellari direttamente nel database si pagano sia **in termini di tempo che in termini di futura stabilità** e linearità dell'applicazione.

Normalizzazione dei Dati

- La **teoria della normalizzazione** ha come scopo quello di fornire metodi per progettare basi di dati senza anomalie.
- Consente di verificare se la definizione dello schema corrisponde a dei "canoni standard" di correttezza della base di dati.
- Dopo aver definito lo schema, si devono seguire alcune regole per rendere le tabelle in quelle che sono chiamate le **FORME NORMALI**, cioè per fare in modo che lo schema corrisponda ai "canoni standard":
  - Ogni tabella deve avere una chiave primaria
  - Ogni campo deve contenere un solo valore (prima forma - forma atomica) **1°**;
  - I campi di una tabella non devono dipendere da altri campi (esclusa la chiave primaria, seconda forma, non ammette ad esempio campi calcolati ottenuti con le query) **2°**;
  - Evitare le ripetizioni e la ridondanza dei dati (terza forma - impostare una relazione 1 a N se necessario) **3°**.

Esempi Normalizzazione

IdOrdine	Data	Quantità	Prezzo	IdCliente	Pagamento
1	21/12/02	100	L. 50.000	1	contanti
2	21/12/02	150	L. 35.000	2	contanti
3	21/12/02	100	L. 35.000	5	rateale
4	21/12/02	50	L. 50.000	25	bonifico
5	30/12/02	100	L. 50.000	1	contanti

Chiave Primaria

Errato

IdPersona	Cognome	Nome	Indirizzo
1	Rossi	Mario	via Roma, 4 - PADOVA
2	Bianchi	Marco	via Trieste, 12 - TREVISO
3	Verdi	Mara	via Aurora, 32 - VENEZIA

Valori multipli in un campo singolo

Corretto

IdPersona	Cognome	Nome	Indirizzo	Nome	Numero	Città
1	Rossi	Mario	via	Roma,	4	PADOVA
2	Bianchi	Marco	via	Trieste	12	TREVISO
3	Verdi	Mara	piazza	Aurora	32	VENEZIA

1°

2°

Errato

IdOrdine	Data	Quantità	PrezzoUnitario	TotaleFattura	Cliente
1	21/12/02	10	€ 5.000,00	€ 50.000,00	23
2	23/12/02	20	€ 7.000,00	€ 140.000,00	14
3	27/12/02	30	€ 6.500,00	€ 195.000,00	37

Non dipende dalla Chiave Primaria

(Va eliminato, lo si può ottenere con Query di Calcolo)

IdOrdine	Data	Quantità	PrezzoUnitario	Cliente
1	21/12/02	10	€ 5.000,00	23
2	23/12/02	20	€ 7.000,00	14
3	27/12/02	30	€ 6.500,00	37

Cliente	Indirizzo	Citta	DataOrdin	Importo
Rossi	Via Roma, 4	Lecce	12-lug	€ [illegible]
Rossi	Via Roma, 4	Lecce	15-ago	€ 5.478,00
Rossi	Via Roma, 4	Lecce	17-ago	€ 4.700,00

Valori Ripetuti

Clienti

IdCliente	Ditta	Indirizzo	Città
1	Rossi	via Roma, 4	Lecce

Ordini

IdOrdine	DataOrdine	Ordine	IdCliente
1	12-lug	7.500.000	1
2	15-ago	5.478.000	1
3	17-ago	4.700.000	1
4	12-set	7.500.000	1
5	15-ott	5.478.000	1

Creare due Tabelle Relazionate 1 a N

3°

Access 97

Microsoft Access 97

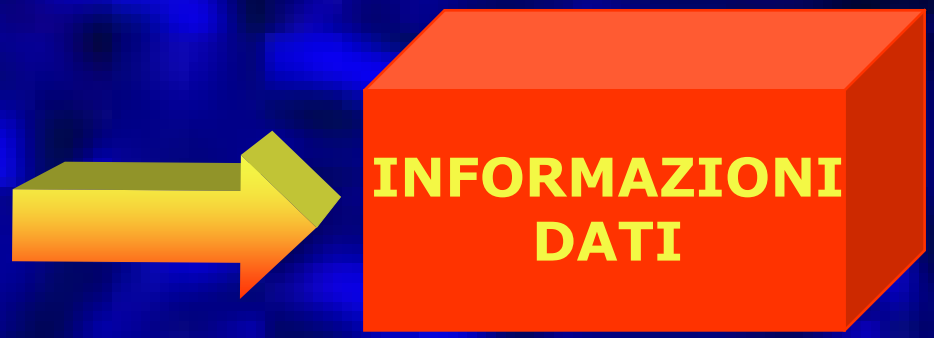
- **Versioni**
- **Caratteristiche**
- **Potenzialità**
- **Interfaccia**
- **Novità**
- **Limiti**

Approfondimenti

Introduzione ai nuovi elementi di Access, autocomposizioni, strumenti di analisi e sicurezza dei Database

Access 97 consente di

Raccogliere
Organizzare
Ricercare
Interrogare
Estrarre
Visualizzare
Stampare
Conservare



Strutturazione Database

- **Elementi fisici del database**
  - **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

Elementi di Access

- Tabelle: rappresentano il contenitori per i diversi record. Sono essenziali per la costruzione del database
- 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
- 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
- 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
- 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.

Elementi di Access

Tabelle → vanno progettato con precisione e lungimiranza. Occorre scegliere i giusti vincoli e le giuste proprietà da assegnargli. Rappresentano il punto di partenza di ogni Database, di cui sono i contenitori dei dati

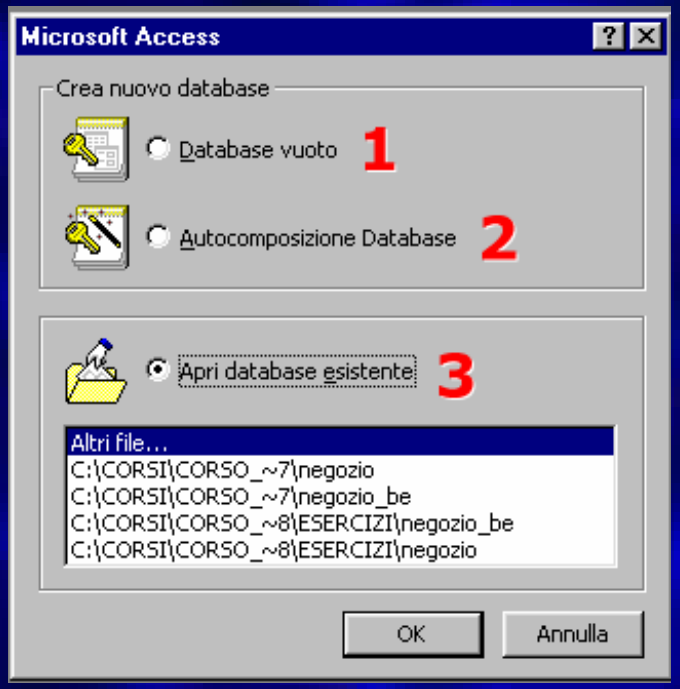
Maschere→ sono essenziali per la gestione, l'inserimento e la visione elaborata e vincolata dei dati. Possiamo dire che rappresentano il metodo con cui Access presenta i dati contenuti nelle tabelle

Query→ consento di filtrare ed elaborare i dati, trasformando e aggiungendo nuovi campi (campi calcolati) originati dalla tabelle stesse

Report→ permettono l'elaborazione e la presentazione dei dati su carta

Macro e Moduli→potenziano le automazioni del database consentendo di realizzare vere e proprie applicazioni e funzioni di supporto.

Creazione di un Database



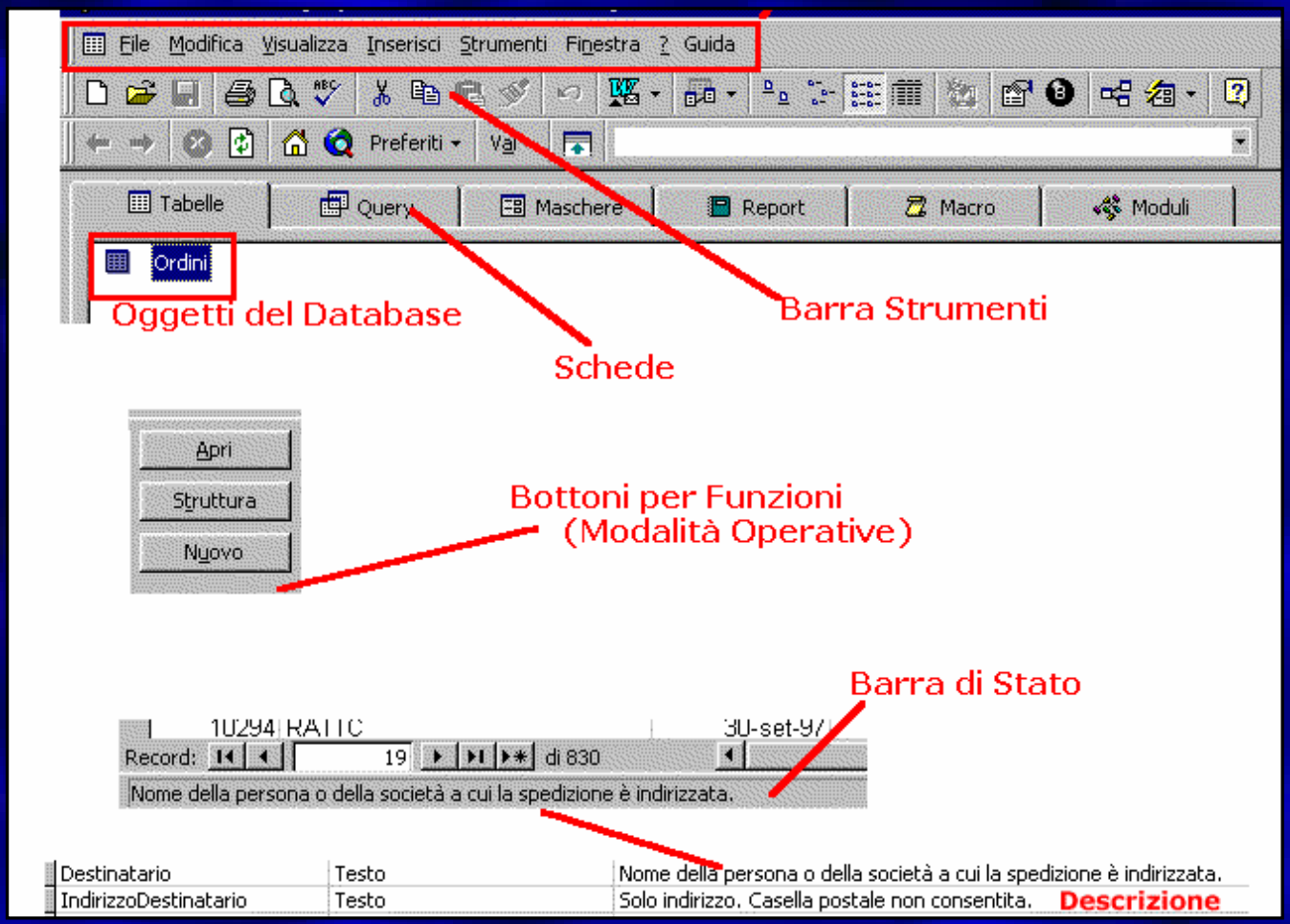
- **Un database si crea (database vuoto - 1), si autocompone (2) o se ne modifica uno esistente (3)**
- **Si può gestire un database per volta**

Un Database di ACCESS

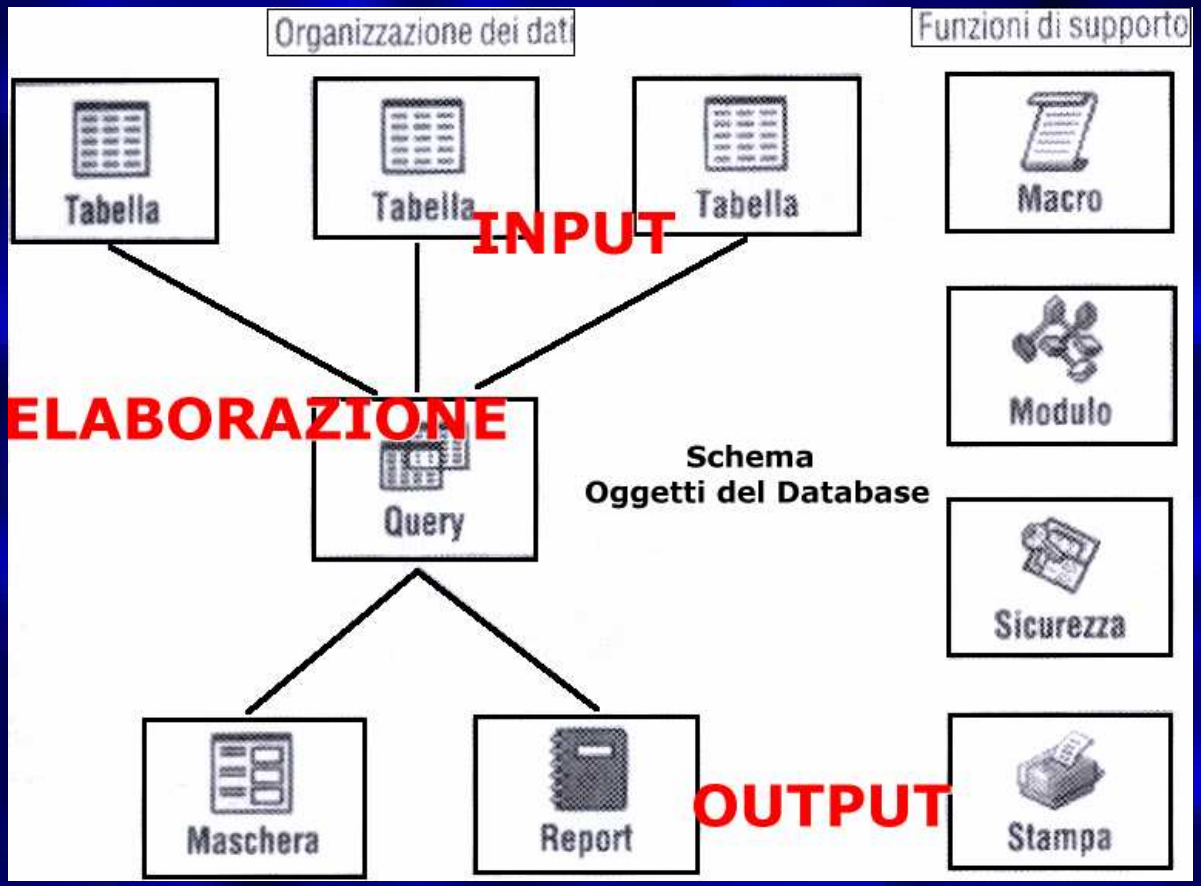
Visualizzazioni

ELEMENTI	VISUALIZZAZIONI	VISUALIZZAZIONI	VISUALIZZAZIONI
Tabelle	Struttura	Fogli Dati	
Maschere	Maschera	Struttura	Fogli Dati
Query	Struttura	SQL	Fogli Dati
Report	Struttura	Anteprima Stampa	
Macro	Struttura		

Interfaccia di ACCESS



Oggetti del Database di Access



Funzionalità di Access

- Access consente la costruzione di database (archivio, contenitore di dati) e di applicazioni complesse
- Access permette di realizzare applicazioni file server, cioè separando l'applicativo dai dati si possono concepire applicazioni distribuite e condivise tra più utenti attraverso una rete.

Le TABELLE in Access

- Sono le unità base di un DB
- Sono raccolte di dati simili
- Le colonne sono identificate dai *campi*, mentre le righe dai *valori* memorizzati negli stessi campi (record).

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	Gamba tavolo 40 cm	[illegible]	[illegible]
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 6(	30000	0050
*			

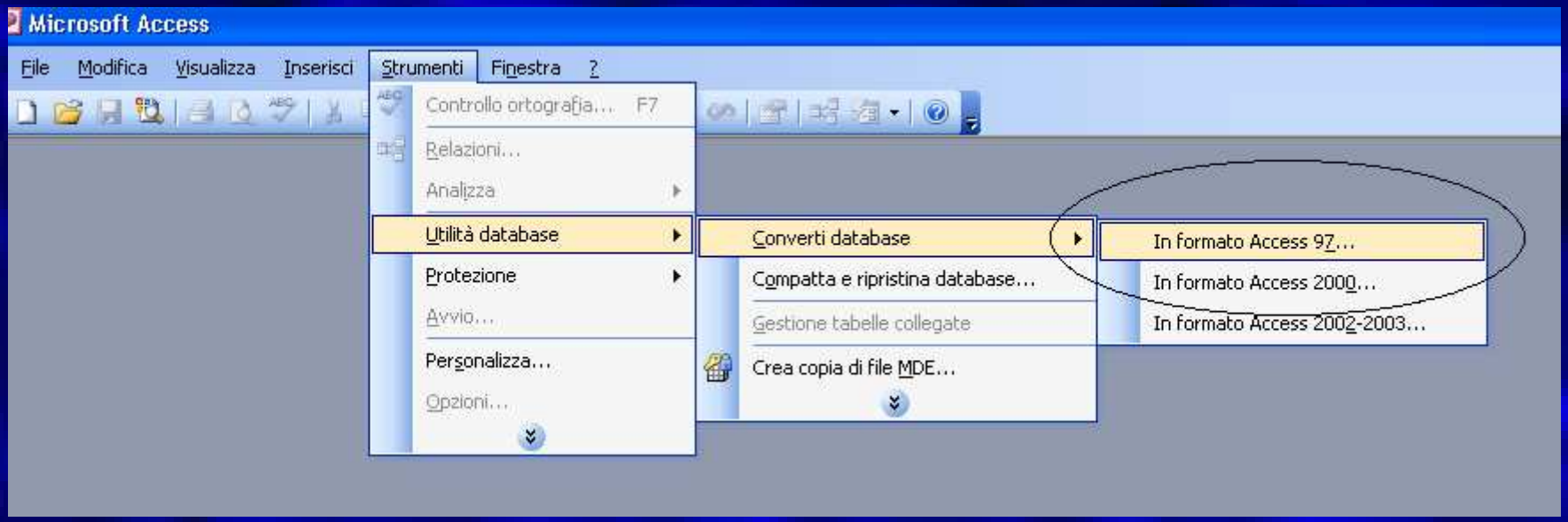
Campo

Record

Estensioni di Access

- .mdb 
- .mde 

I Formati di Access



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 fatto a DB chiuso e va scelta la versione verso cui effettuare la conversione (back port)
Attualmente sono 3:

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

I Formati di Access

- **L'apertura di un DB salvato per una versione <u>più recente</u> con una versione antecedente è impossibile**
- **L'apertura di un DB salvata con una versione <u>meno recente</u> richiede o l'apertura (consigliata solo per operazioni di lettura) o la conversione (consigliata per eseguire modifiche).**

Puntatori di Access

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.

La Tastiera di 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

Icone di Access nel foglio Dati

Simboli per i Record in inserimento

- *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

Interpretazione dei "segnali" di Access

- ***Errori**:* Access dialoga con l'utente segnalando i problemi dovuti ad errori di inserimento o di impostazione: Leggere con attenzione gli errori per capire dove è situato il problema

- ***Blocchi**:* impediscono di violare le condizioni impostate. Access spesso impedisce di proseguire proprio perché dei vincoli sono stati impostati. Occorre rispettare quei vincoli se si vogliono, ad esempio, inserire o modificare dei dati.

Termini per le Tabelle

Tabella, Entità, Relazione

Microsoft Access - [Prodotti : Tabella]

File Modifica Visualizza Inserisci Formato Record Strumenti Finestra ?

ID prodotto	Nome prodotto	Categoria	Quantità per unità	Prezzo unitario
1	Chai	Condimenti	10 scatole da 20 filtri	L. 27.000
2	Chang	Bevande	24 bottiglie da 360 ml	L. 28.500
3	Aniseed Syrup	Bevande	12 bottiglie da 550 ml	L. 15.000
4	Chef Anton's Cajun Seasoning	Bevande	48 vasetti da 180 g	L. 33.000
5	Chef Anton's Gumbon Mix	Prod. ittici	36 scatole	L. 32.025
6	Grandma's Boysenberry Spread	Condimenti	12 vasetti da 250 g	L. 37.500
7	Uncle Bob's Organic Dried Pears	Prod. agricoli	12 confezioni da 300 g	L. 45.000
8	Northwoods Cranberry Sauce	Condimenti	12 vasetti 360 g	L. 60.000

Record, Riga, Istanza, Tupla

Campo, Attributo, Colonna

Relazione, Associazione, Legame

Microsoft Access - [Categorie : Tabella]

File Modifica Visualizza Inserisci Formato Record Strumenti Finestra ?

ID categoria	Nome categoria	Descri
1	Bevande	Bibite analcoliche, caffè, tè e bir
2	Condimenti	Salse dolci e piccanti, sottaceti,
3	Dolciumi	Dessert, caramelle e dolci
4	Latticini	Formaggi
5	Cereali	Pane, cracker, pasta e cereali
6	Carne/pollame	Carni conservate
7	Prod. agricoli	Frutta secca e formaggio di soia
8	Prod. ittici	Alghe e pesce

Struttura di una TABELLA

- **Elementi essenziali nella struttura di TABELLA:**
  - Nome dei campi
  - Tipo di campi (Tipologia Dati)
  - Chiave primaria e indici

N.B: Per poter modificare le caratteristiche di un campo di una determinata tabella occorre disabilitare le eventuali relazioni esistenti

Caratteristiche dei Campi (Tipi di Dati)

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

Codice ASCII e UNICODE

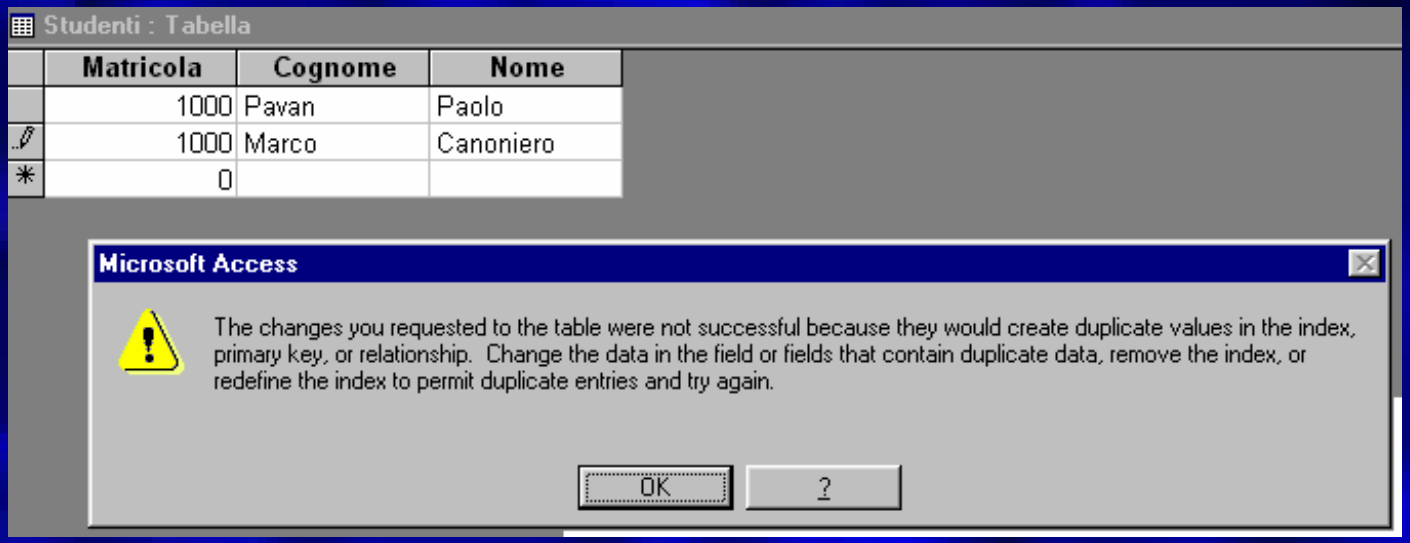
- E' una proprietà, introdotta con la versione 2000 e non presente nelle precedenti, riguarda solamente il campo testo.
- Per rappresentare i caratteri, si utilizzava in maniera esclusiva, fino a qualche anno fa, il **codice ASCII,** un codice standard americano che utilizza un byte per ogni carattere, quindi **8 bit** (vale a dire 2^8 = 256 simboli diversi). Come si è visto sopra, con 8 bit è possibile memorizzare al massimo 256 simboli diversi. Il codice ASCII rappresenta tutti i simboli della tastiera, in più tutti i possibili caratteri americani ed europei, nonché alcuni simboli particolari.
  Il problema del codice ASCII è che, con 256 simboli, non è in grado di rappresentare tutti i simboli dell'alfabeto mondiale, come per esempio i caratteri cinesi e giapponesi.
- Si è dovuto, perciò, definire un nuovo codice, **l'UNICODE (**codice unificato). L'Unicode utilizza 2 byte per ogni carattere, cioè **16 bit** (vale a dire 2^{16} = 65.536 simboli diversi: in tale modo è possibile rappresentare tutti i caratteri).

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 tabelle 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**.

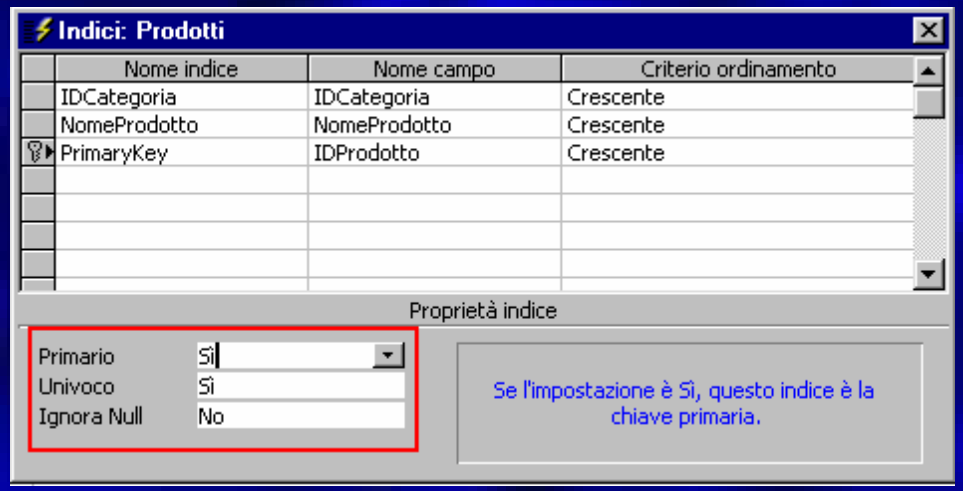
Access e la Chiave Primaria



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

Indice

- Associare il valore indice ad un campo farà si che questo venga indicizzato
  - Un indice può essere composto da uno o più campi
  - Conviene non indicizzare troppi campi pena il rallentamento dell'inserimento dei dati



- 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

Come funziona l'Indicizzazione

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	Bianc		
8	Bianc		
9	Rosa		
10	Gialli		

Indice su Cognome	
Bianchi	7, 8, 13, 18, 24, 26, 34
Gialli	4, 10, 20, 27, 28, 31, 33, 36
Rosa	5, 9, 12, 17, 19, 21, 35
Rossi	2, 11, 15, 16, 22, 25, 30
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**.

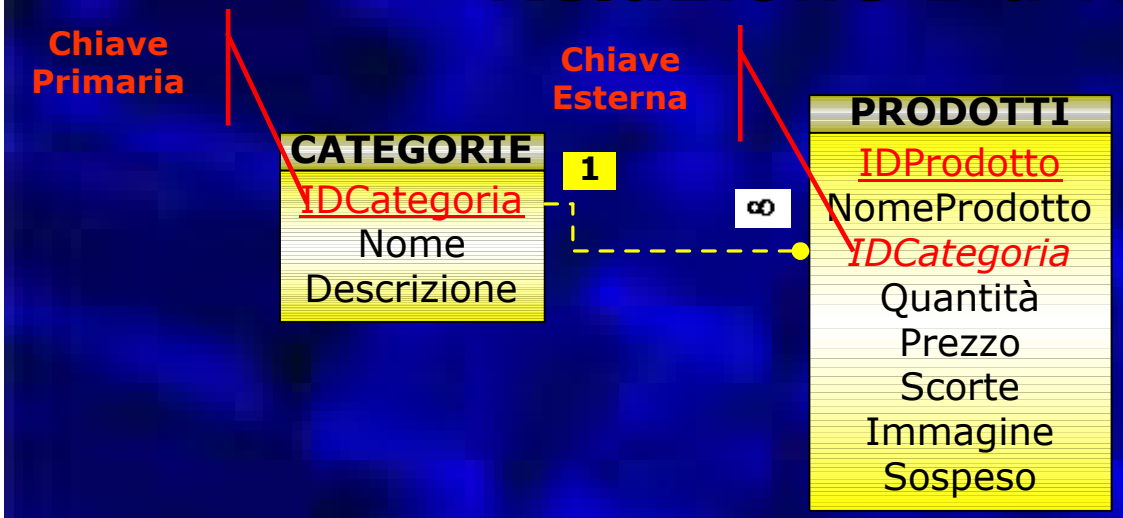
Ad esempio si pensi all'elenco telefonico. La ricerca di un numero sull'elenco del telefono è molto simile a quella su un indice. La ricerca sulla tabella, invece, è simile a quella che si dovrebbe fare se l'elenco telefonico non fosse ordinato, ma i numeri fossero sistemati casualmente. Il paragone 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 del 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

Le RELAZIONI

- Esprime il legame tra due insiemi di dati (tabelle).
- I concetti essenziali sono:
  - Chiave primaria
  - Chiave esterna
  - Relazione 1:1, 1:N;N:N
  - Tipologie di Join (legami)

Esempio di Relazione
Relazione 1 a N (Molti)



Per impostare una relazione 1 a N bisogna aggiungere il campo di chiave primaria del lato uno alla tabella del lato molti, il quale diventa la sua chiave esterna.

- La tabella Categorie intrattiene una relazione del tipo **uno a molti** con la tabella Prodotti:

  "Questo perché un prodotto può appartenere ad un unica categoria mentre una categoria può contenere più prodotti diversi".

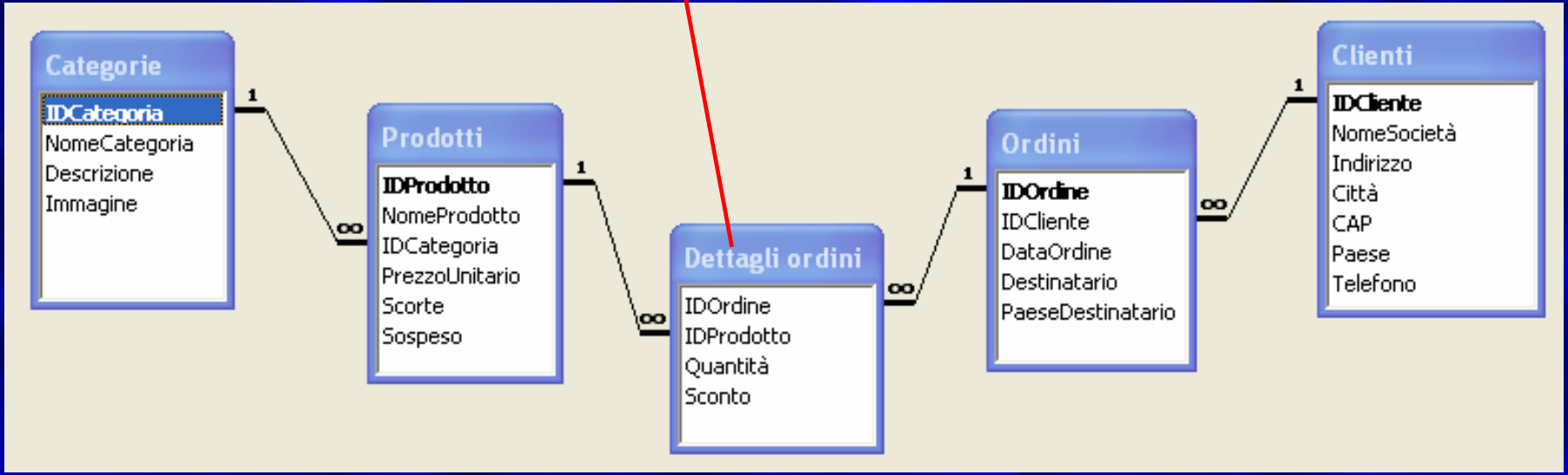
  In parole povere una determinata categoria appare una sola volta nella tabella **Categorie** mentre può apparire N volte nella tabella **Prodotti**.
- Le chiavi primarie sono rappresentate dai rispettivi codici (IDCategorie e IDProdotti) mentre la *chiave esterna* in italico, nella tabella Prodotti rappresenta il campo che consente di collegare le due tabelle.
- La rappresentazione di questa semplice struttura dati su di una singola tabella, sarebbe stata impossibile, in quanto ci avrebbe costretti a duplicare dati (cosa che viola le regole normali), in quanto per ogni prodotto della stessa categoria il codice della categoria sarebbe stato ripetuto, cosa impossibile visto che si tratta della chiave primaria della tabella, **ovvero il campo che deve essere sempre differente, in quanto garantisce l'unicità di ogni singolo record.**

Esempio di Relazione
Relazioni Multiple

Tabella creata come soluzione di una relazione del tipo N a N

Schema completo per la gestione di un Negozio

Prodotti e Categorie
Ordini e Clienti



Esempio di Relazione
Il caso di una Relazione N a N

TStudenti

Matricola	Cognome	Nome
78952	Rossi	Armando
78953	Bianchi	Ginetto
78954	Verdi	Rosita

Tcorsi

Materia	Insegnante	TotaleOre
Basi di Dati	Risi Gino	50
Analisi I	Giallo Marco	60
Sistemi operativi	Risi Gino	50

TStudenti

Matricola	Cognome	Nome	Materia
78952	Rossi	Armando	Basi di dati, Analisi I
78953	Bianchi	Ginetto	Analisi I, Sistemi operativi
78954	Verdi	Rosita	Sistemi operativi, basi di dati

Tcorsi

Materia	Insegnante	TotaleOre
Basi di Dati	Risi Gino	50
Analisi I	Giallo Marco	60
Sistemi operativi	Risi Gino	50

Errori:

Tab up: + valori per un solo campo

Tab down: + valori per un solo campo

TStudenti

Matricola	Cognome	Nome
78952	Rossi	Armando
78953	Bianchi	Ginetto
78954	Verdi	Rosita

Tcorsi

Materia	Insegnante	TotaleOre	Matricola
Basi di Dati	Risi Gino	50	78952,78953
Analisi I	Giallo Marco	60	78953,78954
Sistemi operativi	Risi Gino	50	78954,78952

Soluzione Corretta

2 relazioni 1 a N

Normalizzazione: esercizio
Prima FORMA NORMALE

Clienti

IDCliente	Nome	Indirizzo
1	Paolo	Via Alpignano Grugliasco 5
2	Elena	Cso Francia Torino 67
3	Enrico	Piazza Statuto Milano 23
4	Giuseppe	Via Verdi Viterbo 113
5	Franco	Vicolo Corto Napoli 34

Ogni Record deve avere un valore univoco (chiave Primaria)
Forma Atomica

IDCliente	Nome	Indirizzo	Città	Numero Civico
1	Paolo	Via Alpignano	Grugliasco	5
2	Elena	Cso Francia	Torino	67
3	Enrico	Piazza Statuto	Milano	23
4	Giuseppe	Via Verdi	Viterbo	113
5	Franco	Vicolo Corto	Napoli	34
6	Mario	Via Rossi	Verona	56

Non devono esistere più valori (array) in uno stesso campo: uno campo può contenere un solo valore:
Forma Atomica

Normalizzazione: esercizio
Seconda FORMA NORMALE

Prodotti

IDProdot	Nome	Prezzo	Quantità	Totale
0001	Biscotti	10	3	30
0002	Salumi	8	4	32
0003	Pane	3	5	15
0004	Zucchero	5	5	25

In una tabella **non possono esserci campi calcolati**: Ogni campo deve **dipendere** univocamente **dalla sua chiave primaria**

Il campo Totale dipende dal campo prezzo e dal campo quantità e non dal codice del Prodotto che è la chiave primaria della tabella.
Questo valore **si ottiene con una Query di Calcolo**

Normalizzazione: esercizio
Terza FORMA NORMALE

Se in una tabella si ripetono i valori in diversi campi allora è necessario creare una nuova tabella e **costruire una relazione del tipo** 1 a N (MOLTI)

Ordini

IDOrdine	Data	Cliente	Indirizzo	Città	Nu
1001	10/10/2003	Paolo	Via Alpignano	Grugliasco	
1002	11/06/2003	Elena	Cso Francia	Torino	
1003	12/10/2003	Enrico	Piazza Statuto	Milano	23
1004	13/08/2003	Giuseppe	Via Verdi	Viterbo	113
1006	14/07/2003	Elena	Cso Francia	Torino	56
1007	15/10/2003	Paolo	Via Alpignano	Grugliasco	34

E' essenziale **evitare ripetizioni** nei campi per alleggerire il database e rendere più veloci le operazioni di ricerca e carico dati

Clienti

IDCliente	Nome	Indirizzo	Città	Numero Civico
1	Paolo	Via Alpignano	Grugliasco	5
2	Elena	Cso Francia	Torino	67
3	Enrico	Piazza Statuto	Milano	23
4	Giuseppe	Via Verdi	Viterbo	
5	Franco	Vicolo Corto	Napoli	
6	Mario	Via Rossi	Verona	

Ordini

IDOrdine	Data	Importo	IDCliente
1001	10/10/2003	15,00	1
1002	11/06/2003	10,00	2
1003	12/10/2003	5,00	3
1004	13/08/2003	8,00	4
1006	14/07/2003	7,00	2
1007	15/10/2003	20,00	1

Normalizzazione: esercizio
Relazione N a N (MOLTI a MOLTI)

Prodotti

IDProdotto	Nome	Prezzo	Ordine
0001	Biscotti	10	1001,1002
0002	Salumi	8	1002
0003	Pane	3	1003,1004
0004	Zucchero	5	1001,1003,1004

Ordini

IDOrdine	Cliente	Destinazione	Prodotto
1001	Paolo	Collegno	Biscotti,Zucchero
1002	Elena	Pinerolo	Salumi
1003	Enrico	Milano	Pane,Zucchero
1004	Giuseppe	Roma	Zucchero

Se io volessi associare un prodotto ad un determinato Ordine oppure assegnare ad un Ordine i prodotti Contenuti come dovrei operare?
La creazione di una relazione in questo caso crea una **violazione delle forme normali** in quanto sia aggiungendo gli ordini ai prodotti che i prodotti agli ordini avremo una ripetizione di valori per un singolo campo

La situazione **si risolve creando una terza (nuova**) tabella con cui entrambe le tabelle (Prodotti e Ordini) intrattengono una relazione del tipo 1 a N (MOLTI)

La chiave primaria di questa tabella è rappresentata da **entrambe le chiavi delle tabelle precedenti contemporaneamente**. Entrambe le chiavi possono essere duplicate ma non insieme.

Dett Ordini

IDOrdine	IDProdotto	Prezzo	Quantità
1001	0001	10	3
1001	0004	5	2
1002	0002	8	3
1003	0003	3	1
1003	0004	5	1
1004	0004	5	2

Perché si usano le Relazioni

- Consentono di ridurre la dimensione globale del database evitando ripetizioni
- Consentono una migliore organizzazione e visione del database
- Consentono l'applicazione della logica dell'**Integrità Referenziale.**

Tipi Di JOIN

JOIN tra Tabelle

ID categoria	Nome categoria	Nome prodotto	Prezzo unitario
2	Condimenti	Grandma's Boysenberry Spread	L. 37.500
2	Condimenti	Northwoods Cranberry Sauce	L. 60.000
4	Latticini	Queso Cabrales	L. 31.500
4	Latticini	Queso Manchego La Pastora	L. 57.000
6	Carne/pollame	Chang	L. 28.500
6	Carne/pollame	Chef Anton's Cajun Seasoning	L. 33.000
7	Prod. agricoli	Uncle Bob's Organic Dried Pears	L. 45.000
8	Salumi	Chef Anton's Gumbon Mix	L. 32.025
(Contatore)			

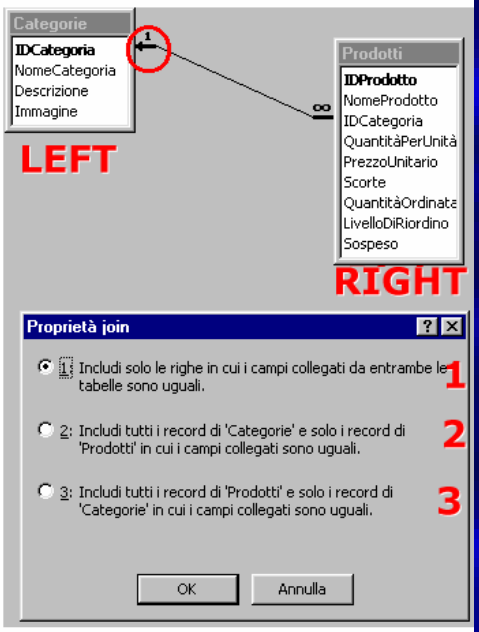
1

ID categoria	Nome categoria	Nome prodotto	Prezzo unitario
1	Bevande		
2	Condimenti	Grandma's Boysenberry Spread	L. 37.500
2	Condimenti	Northwoods Cranberry Sauce	L. 60.000
3	Dolciumi		
4	Latticini	Queso Cabrales	L. 31.500
4	Latticini	Queso Manchego La Pastora	L. 57.000
5	Cereali		
6	Carne/pollame	Chang	L. 28.500
6	Carne/pollame	Chef Anton's Cajun Seasoning	L. 33.000
7	Prod. agricoli	Uncle Bob's Organic Dried Pears	L. 45.000
8	Salumi	Chef Anton's Gumbon Mix	L. 32.025
(Contatore)			

2

ID categoria	Nome categoria	Nome prodotto	Prezzo unitario
		Chai	L. 27.000
		Aniseed Syrup	L. 15.000
		Mishi Kobe Niku	L. 145.500
		Ikura	L. 46.500
2	Condimenti	Grandma's Boysenberry Spread	L. 37.500
2	Condimenti	Northwoods Cranberry Sauce	L. 60.000
4	Latticini	Queso Cabrales	L. 31.500
4	Latticini	Queso Manchego La Pastora	L. 57.000
6	Carne/pollame	Chang	L. 28.500
6	Carne/pollame	Chef Anton's Cajun Seasoning	L. 33.000
7	Prod. agricoli	Uncle Bob's Organic Dried Pears	L. 45.000
8	Salumi	Chef Anton's Gumbon Mix	L. 32.025
(Contatore)			

3



INNER JOIN → JOIN INTERNO (EQUI JOIN)
OUTER JOIN → JOIN ESTERNO (RIGHT JOIN e LEFT JOIN)

Query su più Tabelle

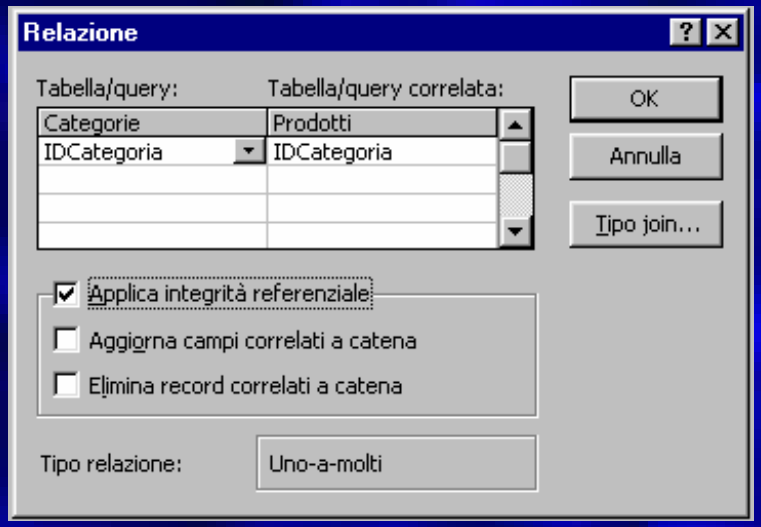
*L'*uso dei Join consente di effettuare interrogazioni che danno esisti differenti su tabelle relazionate.
L'esempio prevede due tabelle di un database di un negozio, in cui in una si schedano i prodotti nell'altra le categorie in cui si ordinano i prodotti.
Le tre differenti applicazioni del join hanno il seguente effetto

Caso 1 (Equi Join): è l'impostazione di default la query restituisce le righe che devono essere equivalenti (uguali) per entrambe le tabelle. Avremo quindi solo i prodotti inventariati e che quindi ricadono in una ben specifica Categoria.
Caso 2 (Join Esterno/Left Join – freccia verso Tab Prodotti): Vengono visualizzati tutti i prodotti associati ad una categoria e tutte le categorie esistenti anche se non contenenti nessun prodotto.
Caso 3 (Join Esterno/Right Join - freccia verso Tab Categorie): in questo caso vengono visualizzati tutti i prodotti in elenco anche se non associati ad una categoria oltre a tutte le categorie a cui è associato un prodotto.
Se la **freccia indica la tabella Categorie**, vuol dire che mostrerà solo i record effettivamente collegati all'altra tabella e viceversa.

Sintassi SQL per JOIN

- Una query può essere espressa in linguaggio SQL:
  - È più potente e versatile se gestita in questo modo
  - È più complessa e richiede la conoscenza del linguaggio SQL
  - Esempio di query con join in formato SQL:
    - *SELECT DISTINCTROW prodotti.Nomeprodotto, prodotti.Prezzounitario, categorie.Nomecategoria, categorie.Descrizione*
      FROM categorie <u>INNER JOIN</u> prodotti ON categorie.Idcategoria = prodotti.Idcategoria;
    - *SELECT DISTINCTROW prodotti.Nomeprodotto, prodotti.Prezzounitario, categorie.Nomecategoria, categorie.Descrizione*
      FROM categorie <u>RIGHT JOIN</u> prodotti ON categorie.Idcategoria = prodotti.Idcategoria;
    - *SELECT DISTINCTROW prodotti.Nomeprodotto, prodotti.Prezzounitario, categorie.Nomecategoria, categorie.Descrizione*
      FROM categorie <u>LEFT JOIN</u> prodotti ON categorie.Idcategoria = prodotti.Idcategoria;

Impostazione di un JOIN



Applica l'integrità referenziale: fa in modo che le operazioni che si effettuano su di una tabella tengano conto anche dei dati presenti in quelle relazionate. Ad esempio Access potrà impedire la cancellazione di un record per cui esistono ancora dati contenuti in una tabella relazionata e impedirà l'inserimento di elementi in una tabella, se non correlati strettamente ai campi di quella relazionata.

Aggiorna campi correlati a catena: se le tabelle sono relazionate, l'aggiornamento del contenuto di un campo in una tabella implica l'aggiornamento dei campi correlati in altro tabelle. **Access avverte l'utente prima di completare l'operazione.**
Per eliminare dati tra tabelle collegate occorre espressamente indicarlo.

Elimina record correlati a catena: se le tabelle sono relazionate, l'eliminazione di un record in una tabella implica l'eliminazione dei record correlati nelle altre tabelle.
Access avverte l'utente prima di completare l'operazione.
La possibilità di violare l'integrità referenziale, modificando o peggio eliminando dei record che sono correlati tra tabelle (caso Categorie-Prodotti) deve essere espressamente indicate. Access ci avverte comunque che la modifica o l'eliminazione di un dato si ripercuote in maniera definitiva anche sui dati nella tabella correlata.

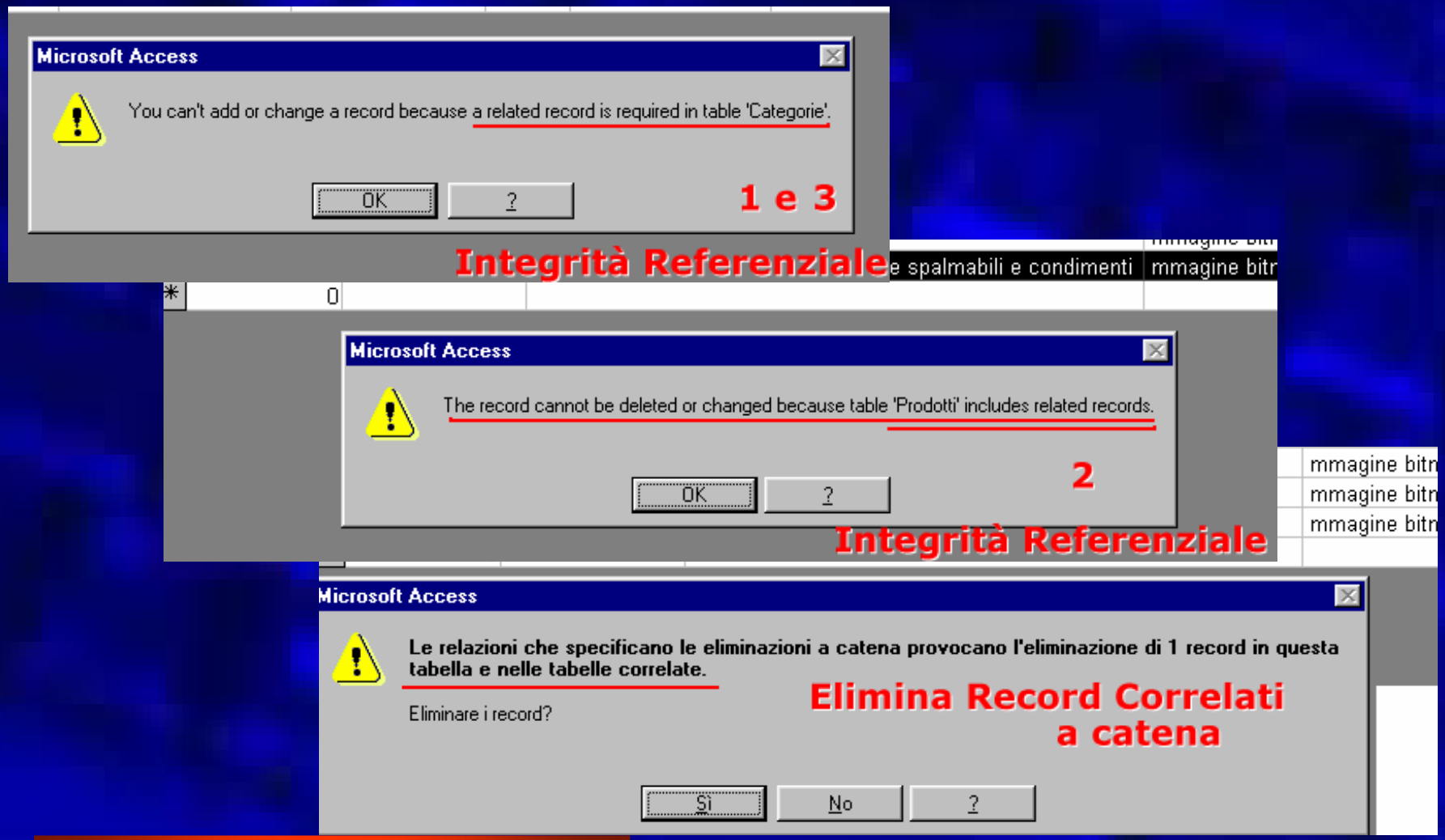
Integrità Referenziale

- L'integrità referenziale, sostanzialmente, effettua un controllo sull'inserimento dei dati: verifica che i valori inseriti nel campo chiave esterna siano "compatibili" con quelli della chiave primaria.

  "A ogni valore non nullo della chiave esterna deve corrispondere una valore della chiave primaria della relazione associata".

1. Non è possibile immettere un valore nella chiave esterna che non esiste nella relazione associata: il valore NULL è ammesso
2. Non è consentito cancellare l'istanza di una chiave primaria se esistono delle istanze uguali nella chiave esterna della relazione associata
3. Non è possibile modificare il valore di una chiave primaria se esistono delle istanze uguali nella chiave esterna della relazione associata.

Effetti dell'integrità Referenziale



Effetti dell'integrità Referenziale

Una volta applicata consente al database di mantenere una coerenza logica:

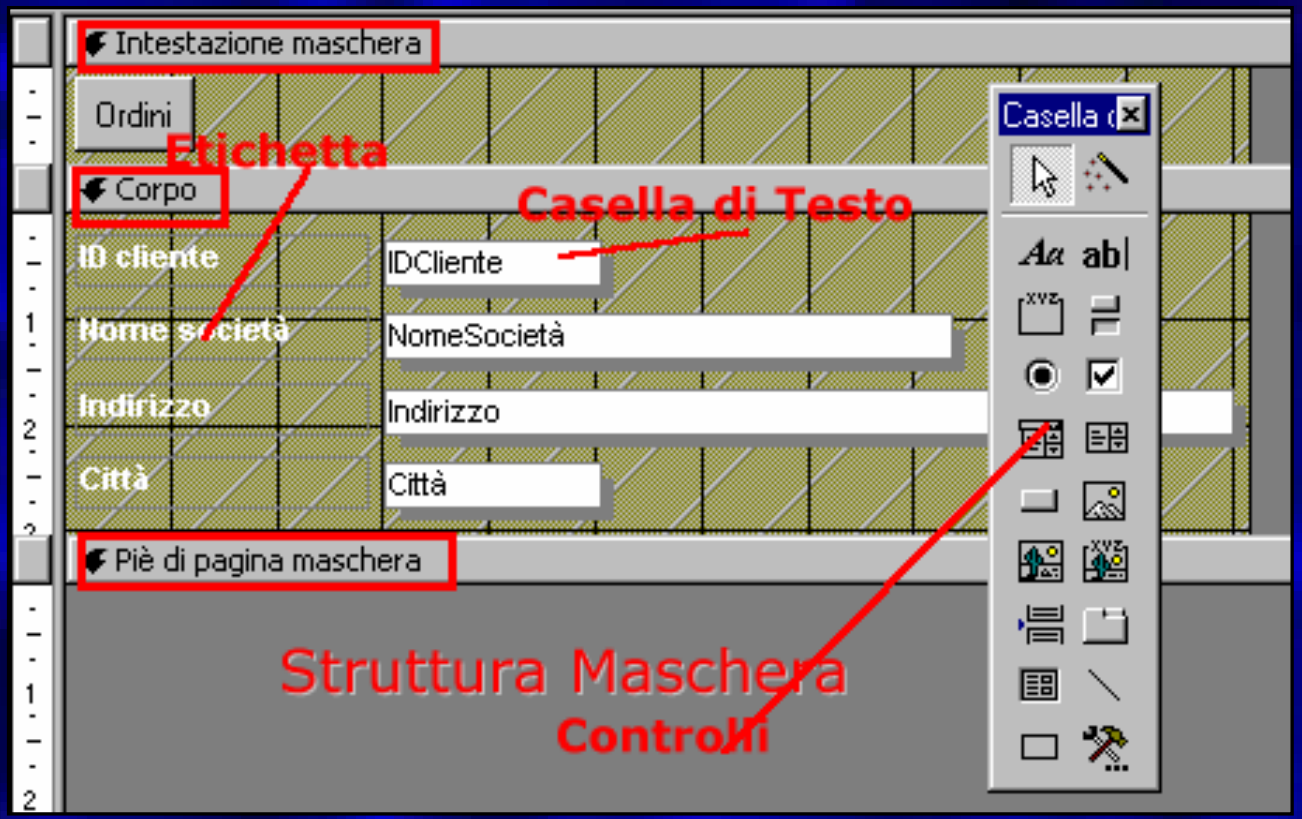
Nell'esempio del Database Negozio applicando l'integrità referenziale non saranno più possibili le seguenti operazioni:

- Eliminare una Categoria se associata a dei Prodotti
- Modificare la chiave primaria (IDCategoria) della tabella Categorie
- Inserire un Prodotto associandolo a una categoria inesistente

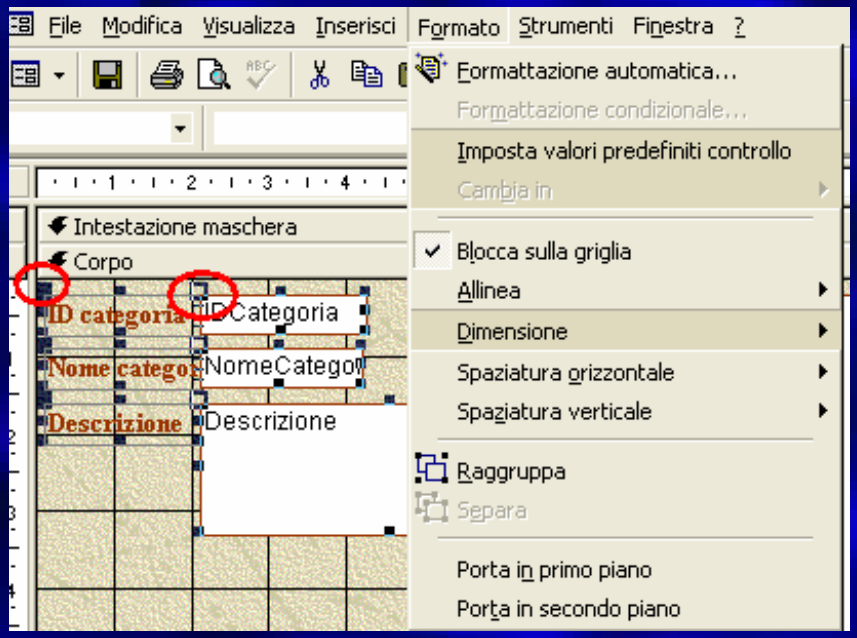
Le MASCHERE in Access

- Rappresentano l'interfaccia principale tra l'utente e l'applicazione di Microsoft Access.
- Hanno svariati utilizzi i principali sono:
  - Controllo del flusso dell'applicazione
  - Accettazione di immissioni
  - Presentazione di messaggi
  - Stampa di informazioni

Struttura di una MASCHERA



Formattazione di una MASCHERA



Formattazione manuale dei controlli:

Allineamento
Spaziatura
Ridimensionamento
Spostamento casella di testo (Manina) o casella di testo+etichetta (Manona)
Raggruppamenti e separazioni

Proprietà dei Controlli

Una maschera possiede delle proprietà specifiche:

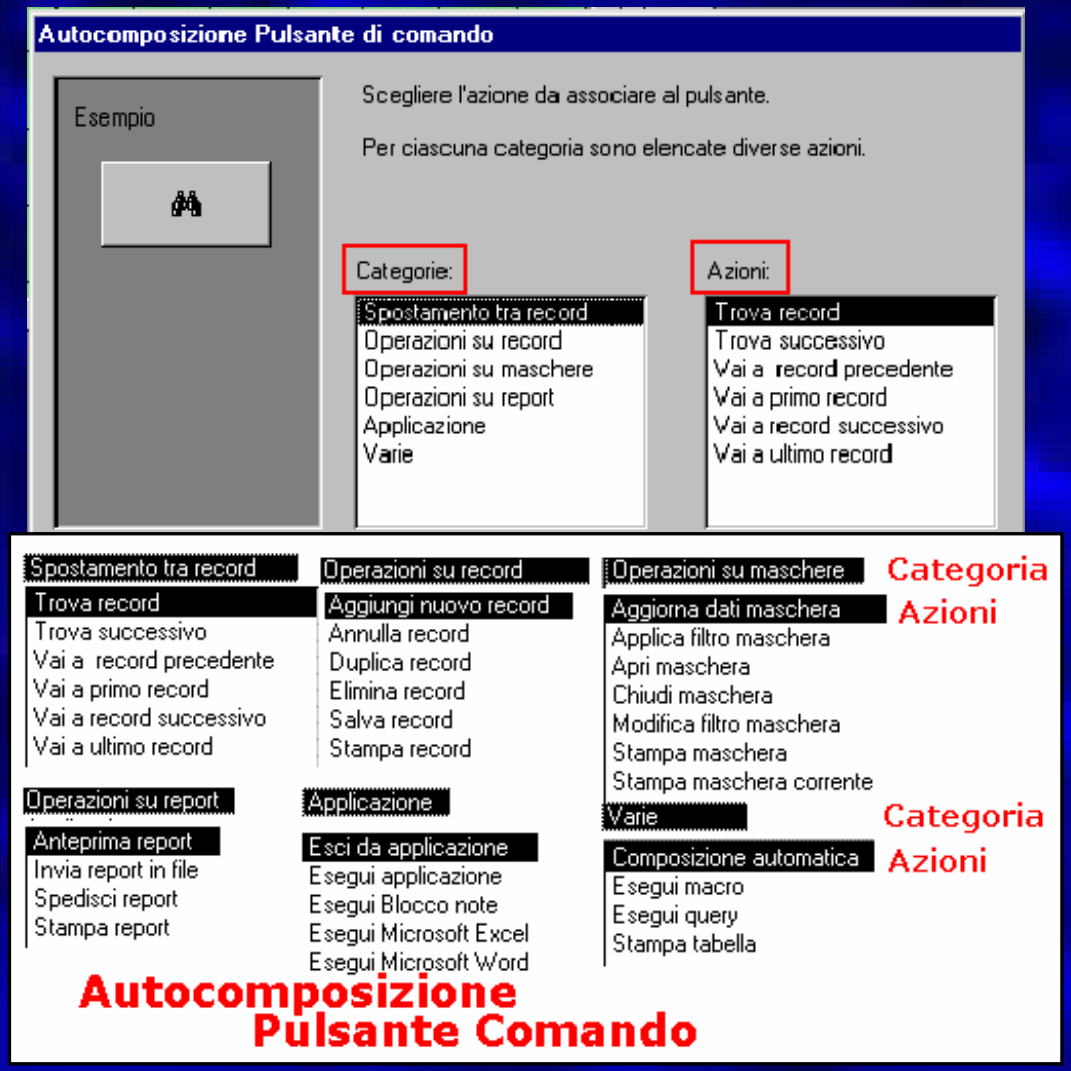
- Proprietà **maschera** (tasto dx su sfondo maschera/Proprietà)
- Proprietà **etichetta** (tasto dx su etichetta/Proprietà)
- Proprietà **casella di testo** (tasto dx su controllo/Proprietà)

Questa finestra controlla ogni parte della casella di testo:

- **Formattazione**: come questa risulta formattata ed appare nella maschera
- **Dati**: qual è l'origine dei dati che visualizza
- **Evento**: indica gli effetti che si ottengono a seconda dell'operazione effettuata sul controllo
- **Altro**: permette di specificare ulteriori opzioni
- **Tutte**: raggruppa in un'unica lunghissima maschere tutte le opzioni

La Barra dei CONTROLLI

Pulsanti di Comando



Consentono **l'automazione delle maschere,** inserendo bottoni che permettono di svolgere differenti operazioni vincolando l'attività degli utenti.

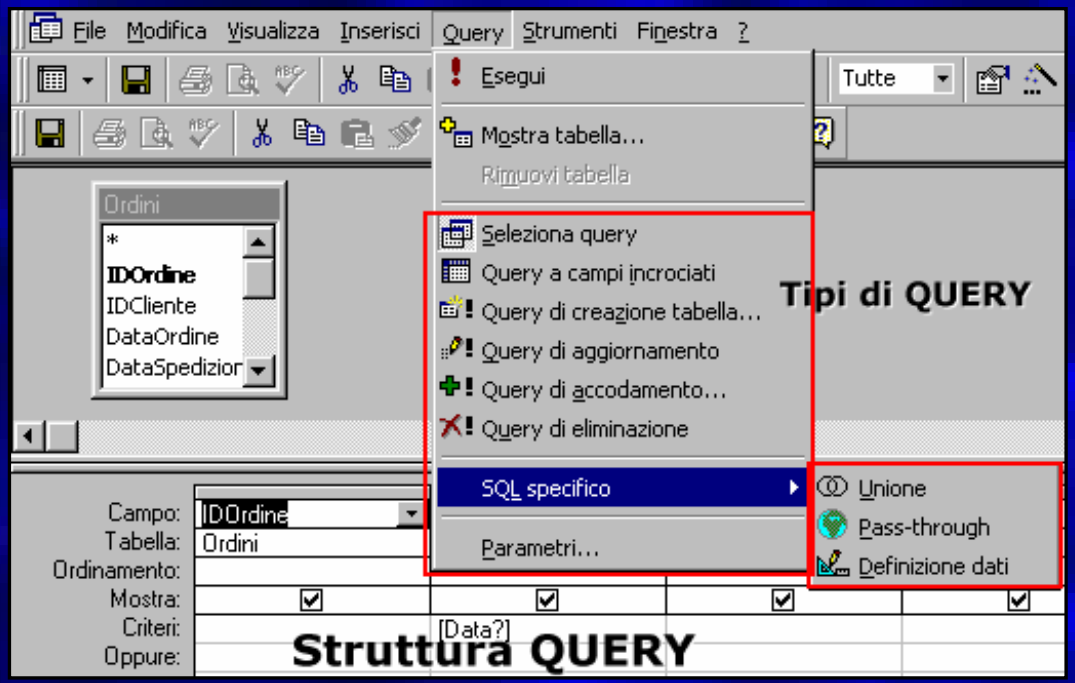
Si suddividono in Categorie ciascuna contenti più Azioni preimpostate facilmente gestibili con comode autocomposizioni.

Ad un bottone è anche possibile associare delle Macro appositamente costruite per superare la rigidità dei delle Azioni preconfezionate.

Le QUERY in Access

- Rappresentano un'interrogazione della base dati.
- Vengono effettuate 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.

Struttura di una QUERY



Campo: nome del campo (basta fare Drag&Drop dalla Tabella)

Tabella: la tabella a cui appartiene il campo

Ordinamento: tipo di ordinamento (crescente o Decrescente)

Mostra: flaggare se il campo coinvolto nella query deve poi effettivamente essere visualizzato

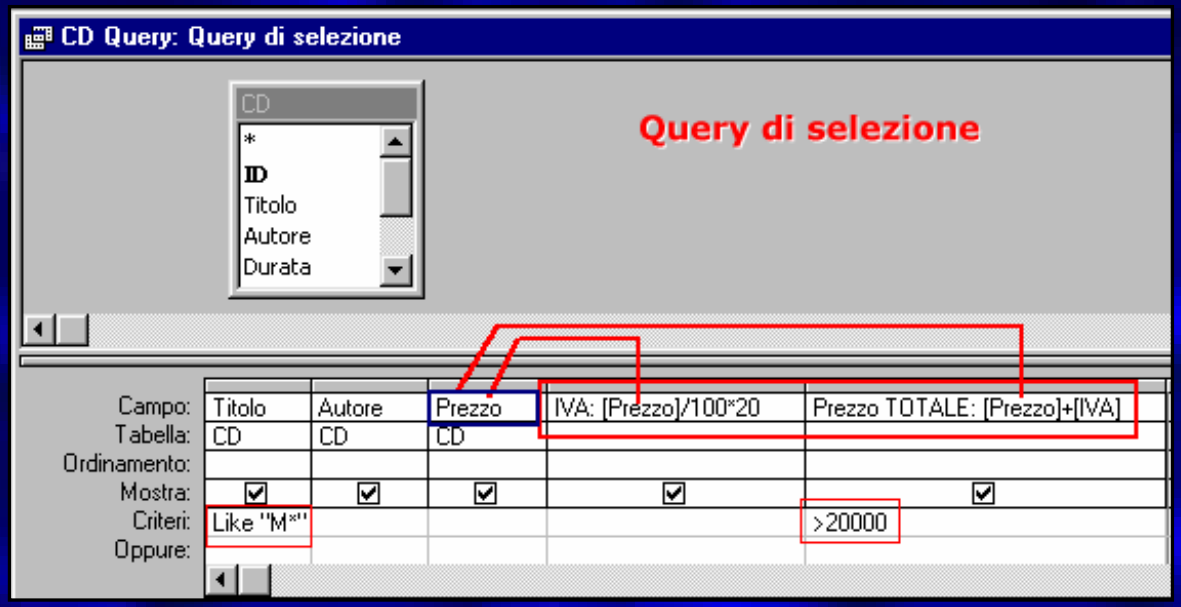
Criteri: impostare i criteri per l'interrogazione

Oppure: consente di impostare un secondo criterio (A AND B)

Tipologie di QUERY

- Query di Selezione
- Query a Campi calcolati
- Query di estrazione Dati
- Query con Formule
- Query per parametri
- Query di comando
- Query a campi incrociati
- Query Ricerca Duplicati
- Query ricerca dati non corrispondenti

Interrogazioni: Le QUERY di Selezione



Elementi in una espressione:

Parentesi quadre ([ ]) per racchiudere nomi di maschera, report, campo o controllo.
Segni di cancelletto (#) per racchiudere date.

Virgolette doppie (") per racchiudere del testo.

Like "Paolo" ricerca il nome paolo
Like "Paol?" ricerca Paolo, paola, paoli
Like "*IT" ricerca tutti i record che finiscono per IT
"02/08/##" o Like "02/08/##" ricerca i record la cui data corrisponde al 2 agosto (# come segnaposto)
"??/??/199?" o Like "??/??/199?" tutti i giorni e mesi degli anni 1990-1999 (? come segnaposto)
#02/02/2000# cancelletti per racchiudere date specifiche
>="M" ricerca tutti i valori che vanno da M a Z
>="#1/1/98" da questa data in poi
Between #2/2/98# AND #1/12/93# tra il 2 febbraio 93 ed il 1 dicembre 93
Between Date () and Date()-90. Date() indica i campi che contengono la data attuale. Con questa espressione si ricerca dalla data di oggi fino a 90 giorni indietro, oppure Between Now() And Now()-90
DatePart→consente di visualizzare la parte di una data
DatePart ("m";[DataOrdine])
DatePart ("aaaa";[DataOrdine]) → queste funzioni non sono criteri e vanno usate nel campo "Campo"
>"n" or <"C" può andare da N a Z oppure da A a C
<>"N" Tutti i valori tranne quelli che iniziano per N
NOT "T*" tutti nomi tranne quelli che iniziano per T

QUERY Risultato - Le Viste

- Una query produce una **vista sui dati** e può essere manipolata come una tabella
- i dati sono modificati sulle tabelle base: <u>modificare le query (vista) vuol dire modificare le tabelle da cui è stata ottenuta</u>
  - Vista = tabella virtuale
  - Tabelle del data base = tabelle primarie
  - Le viste sono finestre dinamiche sulle tabelle del data base : ogni modifica ai dati nella tabella primaria si riflette sulla vista.

Operatori e caratteri universali

OPERATORI DI CONFRONTO			
OPERATORE	SIGNIFICATO	ESEMPIO	SPIEGAZIONE
=	Uguaglianza	= Smith oppure = "Smith"	Uguale a Smith
>	Maggiore di	> 5000	Maggiore di 5000
<	Minore di	<1/1/2000	Minore (precedente) il 1°/1/2000
>=	Maggiore o uguale a	>= M oppure >= "M"	Maggiore o uguale della lettera M (in ordine alfabetico)
<=	Minore o uguale a	<= 31/12/2000 oppure #<= 31/12/2000#	Minore (precedente) o uguale al 31/12/2000
<>	Diverso da	<> CA oppure <> "CA"	Diverso da CA
Between	Tra due valori (inclusi)	Between 15 and 25	Un numero compreso tra 15 e 25 (inclusi)
In	All'interno di un insieme o di un elenco di valori	In [NY AZ NJ] oppure In ["NY", "AZ", "NJ"]	New York, Arizona o New Jersey
Is Null	Il campo è vuoto	Is Null	Record che non hanno alcun valore in questo campo
Is Not Null	Il campo non è vuoto	Is Not Null	Record che hanno un valore in questo campo
Like	Corrisponde a un modello	Like MO-* oppure Like "MO-*"	Record che iniziano con i MO- seguiti da un numero qualsiasi di altri caratteri

OPERATORI LOGICI			
OPERATORE	SIGNIFICATO	ESEMPIO	SPIEGAZIONE
And	Entrambe le espressioni sono vere	<= 1 And <= 10	Tra 1 e 10
Or	È vera una o l'altra delle espressioni, oppure tutte e due	UT Or AZ oppure "UT" Or "AZ"	Utha o Arizona
Not	L'espressione non è vera	Not Like MO-*	Record che non iniziano con i caratteri MO-

CARATTERI UNIVERSALI			
OPERATORE	SIGNIFICATO	ESEMPIO	SPIEGAZIONE
?	Un carattere qualsiasi	P?-100	Valori che iniziano con la lettera P seguita da un carattere qualsiasi, seguito da –100
*	Un numero qualsiasi di caratteri (qualsiasi carattere)	(619)*	Qualsiasi testo che inizia con (619), ad esempio, numeri di telefono o di fax
[nome campo]	Qualche altro campo presente nella griglia QBE	< [Prezzo Unitario]	Record nei quali il valore di questo campo è minore del valore presente nel campo PrezzoUnitario

Operatori di stringa e date

OPERATORI MATEMATICI E DI STRINGA	
OPERATORE	**SIGNIFICATO**
+	Somma
-	Sottrazione
*	Moltiplicazione
/	Divisione
\	Divisione intera
^	Esponenziale
Mod	Resto di divisione (modulo)
&	Concatenazione di due stringhe di testo

Date( ) e DateAdd()	
ESEMPIO	**VALORE RESTITUITO**
Date( )	La data corrente
<= Date( )	La data corrente e tutte le date che la precedono
>= Date( )	La data corrente e tutte le successive
<= Date( ) – 30	Date precedenti a 30 giorni prima della data corrente
Between Date( ) and Date( ) + 30	Date comprese negli ultimi 30 giorni
Between Date( ) and Date( ) - 30	Date comprese nei prossimi 30 giorni
Between Date( ) - 60 and Date( ) - 30	Date comprese tra 30 e 60 giorni fa
> DateAdd("m",1,Date( ))	Date oltre 1 mese ("m") dalla data corrente

QUERY SQL - Operatori Booleani

Criteri di selezione e interrogazione

AND=E

Perché la query abbia effetto devono essere **valide contemporaneamente** tutte e due le condizioni **per lo stesso** record.

*Ad esempio A **AND** B, dice che la query avrà effetto per quei record che contengono **sia A che B.***

OR =O

Perché la query abbia effetto è sufficiente che per ogni singolo record sia valida una condizione o (OR) l'altra, oppure entrambe.

Ad esempio A OR B, dice che la query avrà effetto per quei record che contengono A o B oppure entrambi.

NOT (AND NOT) = E NON

In questo caso la verifica per una condizione è valida solo se l'altra non è verificata.

*Ad esempio A **AND NOT** B, dice che la query avrà effetto solo per quei record che contengono a e **non B.***

Operatori SQL

BETWEEN, IN LIKE

Vengono utilizzati come clausole di predicato utili per specificare particolari intervalli di dati nei recordset.

BETWEEN AND: utile per specificare un intervallo di valori

Esempio: Between 10 AND 20 equivale a specificare >=10 AND <=20, quindi intercetta quei recordset di valori maggiori di 10 ma minori di 20.

IN: utile per specificare un elenco di valori nel quale uno qualsiasi può corrispondere al campo ricercato.

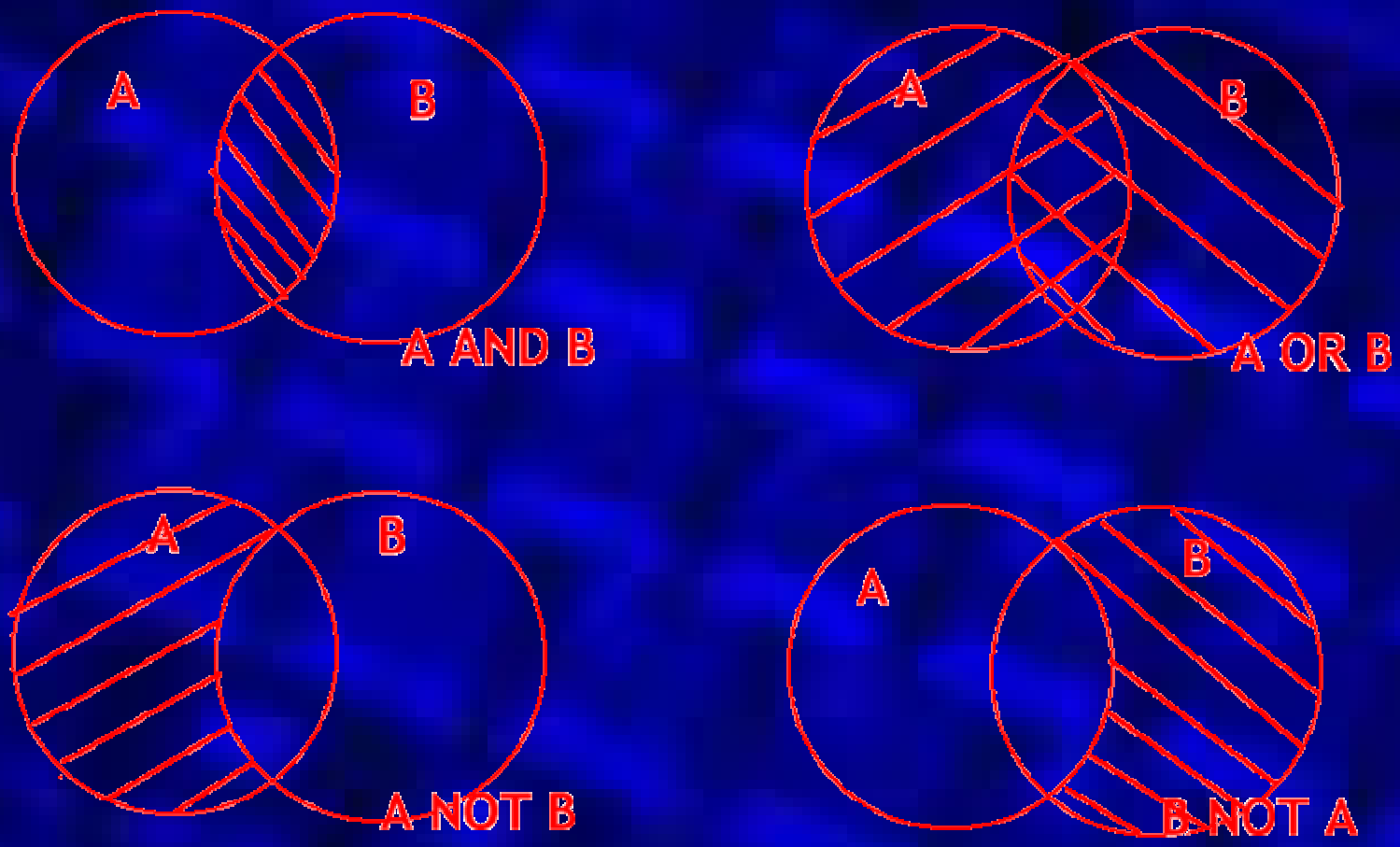
Esempio: IN ("A";"B";"C") equivale a scrivere A OR B OR C

LIKE: molto utile per ricercare schemi nei campi di testo. Infatti LIKE viene utilizzato con i caratteri speciali (*,? E #) per identificare recordset di dati:

Esempio: LIKE A OR LIKE B*, ricerca i dati che per quel campo iniziano con A o B.*

Operatori Booleani

Esempio degli Insiemi



Esempio di Query con Operatori Booleani (AND - OR)

Sintassi SQL

Campo:	NomeProdotto	PrezzoUnitario	QuantitàPerUnità
Tabella:	Prodotti	Prodotti	Prodotti
Ordinamento:			
Mostra:	☑	☑	☑
Criteri:	Like "C*"	>10000	
Oppure:			

AND

```
SELECT Prodotti.NomeProdotto, Prodotti.PrezzoUnitario, Prodotti.QuantitàPerUnità, ([PrezzoUnitario]/100)*20 AS IVA, [PrezzoUnitario]+[IVA] AS Prezzo
FROM Prodotti
WHERE (((Prodotti.NomeProdotto) Like "C*") AND ((Prodotti.PrezzoUnitario)>10000));
```

Campo:	NomeProdotto	PrezzoUnitario	QuantitàPerUnità
Tabella:	Prodotti	Prodotti	Prodotti
Ordinamento:			
Mostra:	☑	☑	☑
Criteri:	Like "C*"		
Oppure:		>10000	

OR

```
SELECT Prodotti.NomeProdotto, Prodotti.PrezzoUnitario, Prodotti.QuantitàPerUnità, ([PrezzoUnitario]/100)*20 AS IVA, [PrezzoUnitario]+[IVA] AS Prezzo
FROM Prodotti
WHERE (((Prodotti.NomeProdotto) Like "C*") OR (((Prodotti.PrezzoUnitario)>10000));
```

Esempio di Query con Operatori Booleani (AND - OR)

Campo:	NomeProdotto
Tabella:	Prodotti
Ordinamento:	
Mostra:	☑
Criteri:	Like "C*" AND Like "P*"
Oppure:	

Sbagliato

Campo:	NomeProdotto
Tabella:	Prodotti
Ordinamento:	
Mostra:	☑
Criteri:	Like "C*" Or Like "P*"
Oppure:	

Campo:	NomeProdotto
Tabella:	Prodotti
Ordinamento:	
Mostra:	☑
Criteri:	Like "C*"
Oppure:	Like "P*"

Corretto

QUERY Varie

Campo:	IDCategoria	MediaUnitaria: Prez:	SommaUnitaria: PrezzoUnitario	ConteggioUnitario: PrezzoUnitario
Tabella:	Prodotti	Prodotti	Prodotti	Prodotti
Formula:	Raggruppamento	Media	Somma	Conteggio
Ordinamento:				
Mostra:	☑	☑	☑	☑
Criteri:				
Oppure:				

Query con Forumule

Ordini
*
IDOrdine
IDCliente
DataOrdine
DataSpedizior

1

Query per parametri

Campo:	IDOrdine	DataOrdine	DataSpedizior
Tabella:	Ordini	Ordini	Ordini
rdinamento:			
Mostra:	☑	☑	☑
Criteri:		[Data?]	
Oppure:			

Immissione valore parametro

Data?

08-ago-97

OK Annulla

Prodotti
*
IDProdotto
NomeProdotto
IDCategoria
QuantitàPerU

Query di Comando
(Aggiornamento)

Campo:	NomeProdotto
Tabella:	Prodotti
Aggiorna a:	"CHAI"
Criteri:	"Chai"
Oppure:	

Microsoft Access

Numero di righe che verranno aggiornate: 1.

Se si sceglie Sì, non sarà possibile utilizzare il comando Annulla per annullare le modifiche.
Aggiornare questi record?

Sì No

Query a campi Incrociati

- Sono utili in presenza di dati numerici che possono fornire dei riepiloghi
- E' una query che presenta valori calcolati nelle caselle definite dalle intersezioni di righe e di colonne. Le righe, le colonne e i valori nelle intersezioni sono ottenuti da tabelle e da query.

Il motore del database: MS Jet Database Engine

- Le applicazioni database Access si articolano in diverse componenti software. In particolare si distinguono **uno strato contenente l'interfaccia utente**, che mette in grado l'utilizzatore finale di interagire con l'applicazione, **ed uno strato**, detto motore del database, **che consente il collegamento tra l'applicazione ed i suoi dati**.
- In tal modo si realizza l'indipendenza del database dalla particolare applicazione che vi accede. Ciò rende possibile, ad esempio, ad un foglio Excel di condividere i dati con una applicazione Access.

Servizi di base del DBMS

- Il MS jet fornisce la connessione ai dati attraverso una serie di servizi tipici dei DBMS.
  - Definizione strutture e vincoli d'integrità
  - Archiviazione (metodo ISAM)
  - Manipolazione e reperimento dati (SQL + DAO)
  - Sicurezza (Workgroup Administrator - Wrkadm.exe)
  - Gestione accessi concorrenti (blocco tabella/record)
  - Manutenzione (compattazione e ripristino)

Microsoft Jet SQL

- Il MS Jet SQL è un dialetto del Structured Query Language, linguaggio per le interrogazioni strutturate compatibile con lo standard ANSI 89.
- Il linguaggio SQL è uno standard industriale rivolto esclusivamente alla manipolazione ed amministrazione di database relazionali.
- Nella sua versione standard si presenta come **un linguaggio applicativo non procedurale**: non contiene né strutture di controllo per l'esecuzione del codice né costrutti che consentano la creazione di maschere di immissione dati o report, destinati alla stampa.

SQL in ACCESS

- **Sintassi Standard**

```
CREATE TABLE nome_tabella (
Campo_1 Tipo Dimensioni Not Null Indice Chiave Primaria,
Campo_2,
Campo_N )

SELECT nome_campo.Tabella FROM Tabella WHERE condizione ORDER BY Asc/Desc
INSERT INTO .Tabella VALUES ('val_1','val_2','val_N ');
DELETE FROM .Tabella WHERE nome_campo=valore;
UPDATE  Tabella SET nome_campo='nuovo valore'  where nome_campo=valore
```

SQL in ACCESS: Tabelle

Creazione di una TABELLA

```
CREATE TABLE Impiegati
(
matricola INTEGER  PRIMARY KEY NOT NULL,
nome CHAR(12) NOT NULL,
cognome CHAR(20) NOT NULL,
mansioni CHAR(50)
);
```

Eliminazione di una TABELLA

```
DROP TABLE Impiegati;
```

Modificare le proprietà di un campo

```
ALTER TABLE Impiegati ADD eta char(5);
ALTER TABLE Impiegati DROP COLUMN eta;
ALTER TABLE Impiegati CHANGE età datanascita;
```

SQL in ACCESS: Query selezione

Query di Selezione

```
SELECT Prodotti.NomeProdotto, Prodotti.QuantitàPerUnità, Prodotti.PrezzoUnitario,
    Prodotti.Sospeso
FROM Prodotti;
```

Query di Selezione con condizione

```
SELECT Prodotti.NomeProdotto, Prodotti.QuantitàPerUnità, Prodotti.PrezzoUnitario,
    Prodotti.Sospeso
FROM Prodotti
WHERE (((Prodotti.PrezzoUnitario)>30000));
```

Query con Ordinamento (ORDER BY)

```
SELECT Prodotti.NomeProdotto, Prodotti.QuantitàPerUnità, Prodotti.PrezzoUnitario,
    Prodotti.Sospeso
FROM Prodotti
WHERE (((Prodotti.PrezzoUnitario)>30000))
ORDER BY Prodotti.NomeProdotto;
```

Query con Raggruppamento (GROUP BY)

```
SELECT DISTINCTROW [Categorie].[NomeCategoria], First([Prodotti].[NomeProdotto]) AS [Primo
    Di NomeProdotto], Sum([Prodotti].[PrezzoUnitario]) AS [Somma Di PrezzoUnitario],
    Avg([Prodotti].[PrezzoUnitario]) AS [Media Di PrezzoUnitario], First([Prodotti].[Scorte]) AS
    [Primo Di Scorte]
FROM Categorie INNER JOIN Prodotti ON [Categorie].[IDCategoria] =[Prodotti].[IDCategoria]
GROUP BY [Categorie].[NomeCategoria];
```

SQL in ACCESS: Query su relazioni

Query su relazione (INNER JOIN)

SELECT [Categorie].[NomeCategoria], [Categorie].[Descrizione], [Prodotti].[NomeProdotto], [Prodotti].[PrezzoUnitario], [Prodotti].[Scorte]

FROM Categorie INNER JOIN Prodotti ON [Categorie].[IDCategoria] =[Prodotti].[IDCategoria];

→ i JOIN Interni (detti anche EQUI JOIN) considerano i record delle due tabelle relazionate ogni volta che nel campo camune ad entrambe le tabelle vengono idividuati valori corrispondenti

Query su relazione (OUTER JOIN – LEFT JOIN)

SELECT Categorie.NomeCategoria, Categorie.Descrizione, Prodotti.NomeProdotto, Prodotti.PrezzoUnitario, Prodotti.Scorte

FROM Categorie LEFT JOIN Prodotti ON Categorie.IDCategoria = Prodotti.IDCategoria;

Include tutti i record della tabella Categorie anche se non associati ad alcun rodotto

Query su relazione (OUTER JOIN – LEFT JOIN)

SELECT Categorie.NomeCategoria, Categorie.Descrizione, Prodotti.NomeProdotto, Prodotti.PrezzoUnitario, Prodotti.Scorte

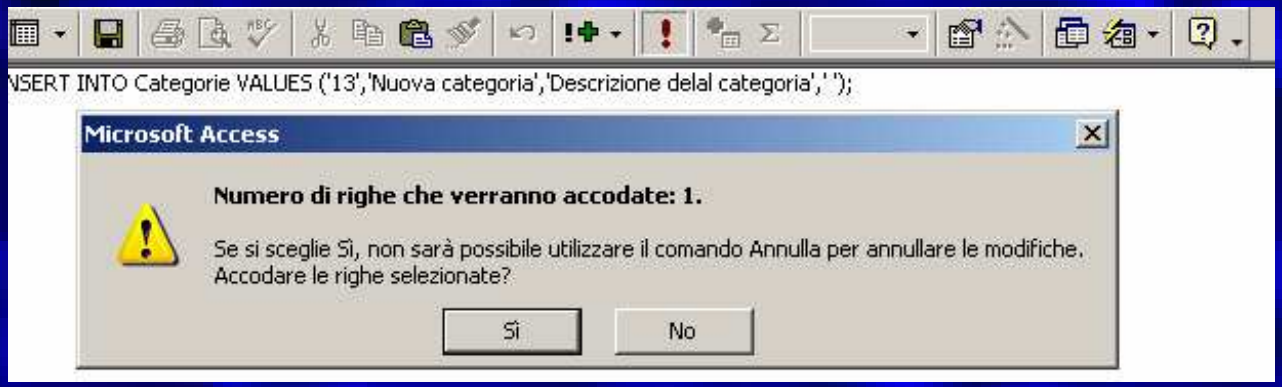
FROM Categorie RIGHT JOIN Prodotti ON Categorie.IDCategoria = Prodotti.IDCategoria;

→ Include tutti i Prodotti anche quelli che non ricadono in nessuna categoria

SQL in ACCESS: Gestione Record

Inserimento RECORD

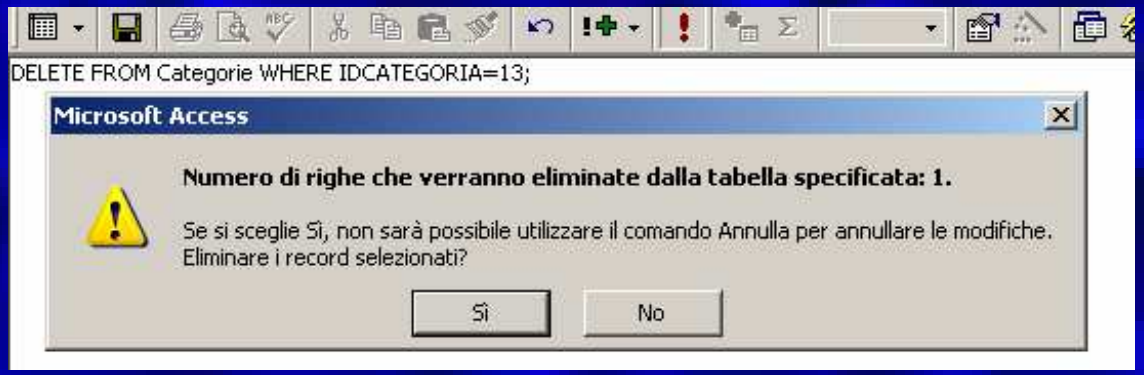
INSERT INTO Categorie VALUES ('13','Nuova categoria','Descrizione delal categoria',' ');



SQL in ACCESS: Gestione Record

Eliminazione RECORD

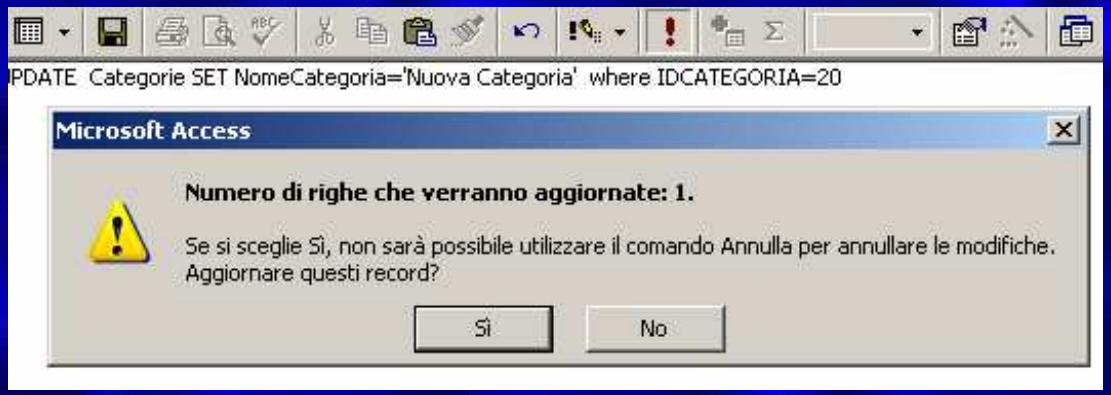
DELETE FROM Categorie WHERE IDCATEGORIA=13;



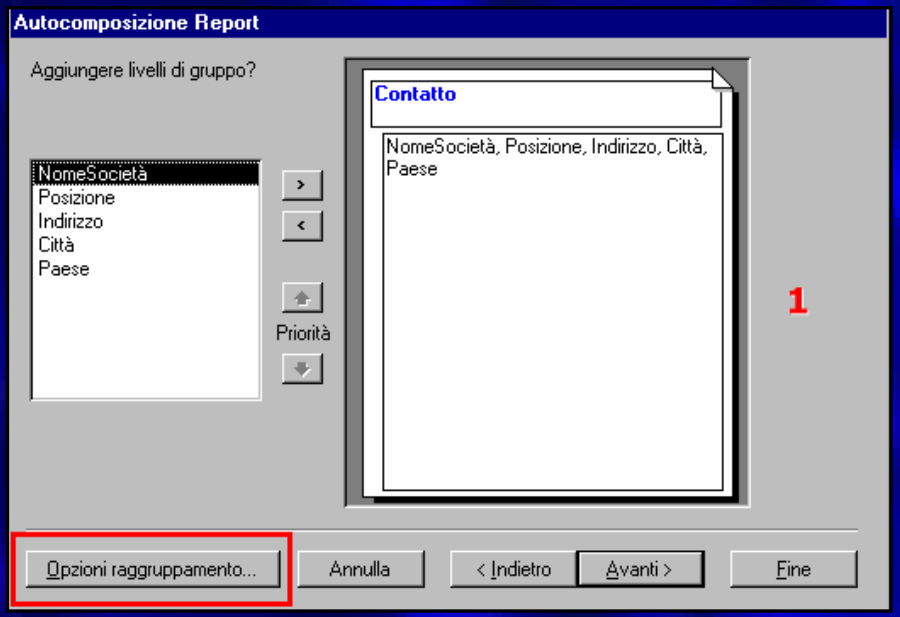
SQL in ACCESS: Gestione Record

Aggiornamento RECORD

UPDATE Categorie SET NomeCategoria='Nuova Categoria' where IDCATEGORIA=20;

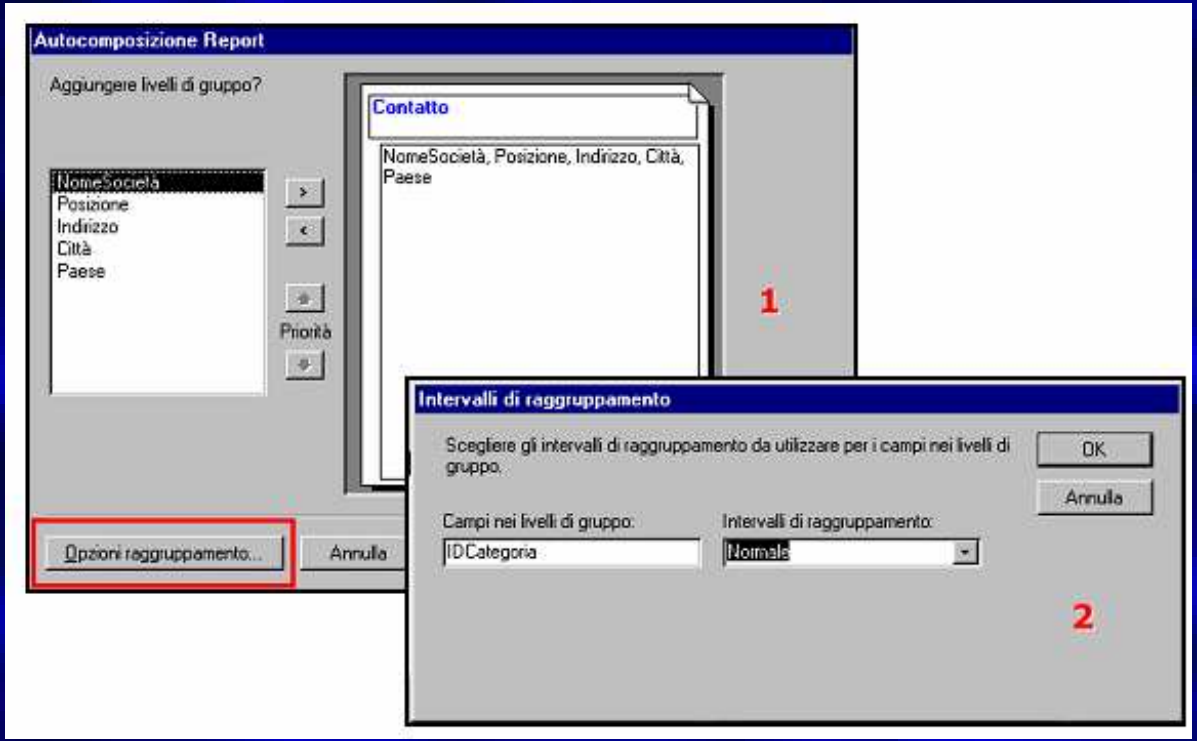


I REPORT in Access



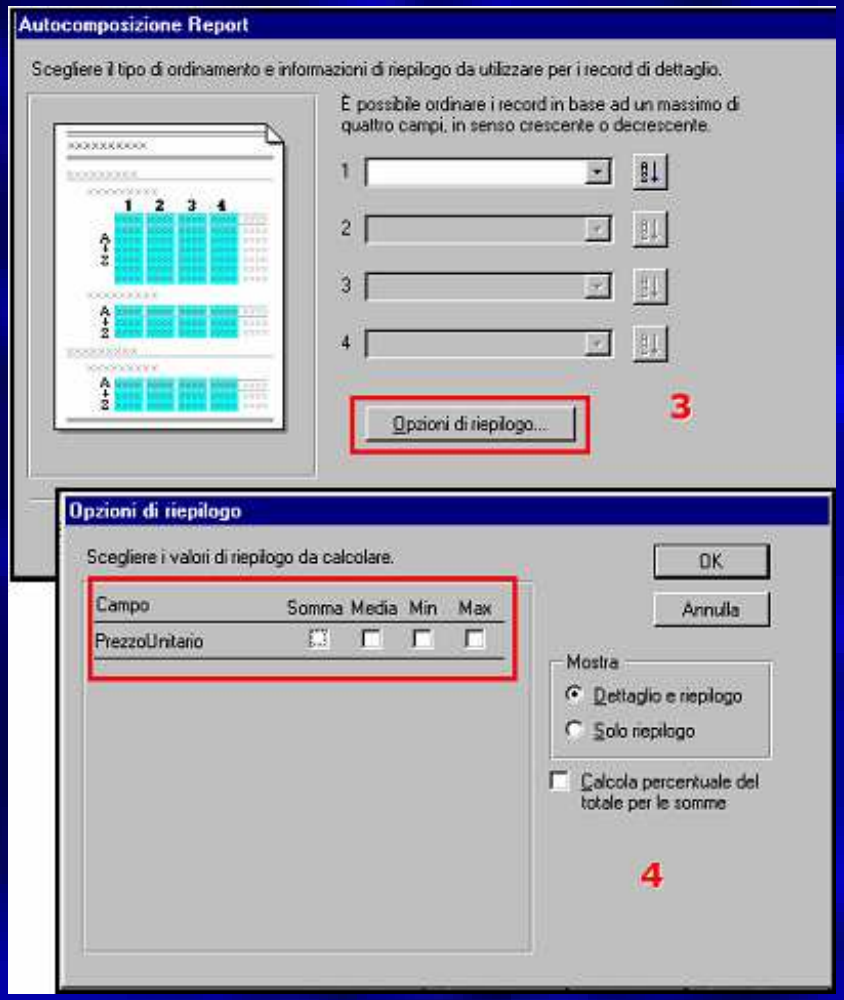
- La loro funzionalità principale e preparare i dati per la stampa. Ovviamente è possibile rielaborare i dati (provenienti da query e tabelle).
- Costituiscono il miglior metodo per creare una copia stampata delle informazioni estratte o calcolate dei dati presenti nel database.

Funzioni dei Report
Raggruppamento e ordinamento



- In presenza di tabelle relazionate o di campi che possono consentire un raggruppamento per una migliore lettura e presentazione dei dati contenuti è possibile stabilire il tipo di raggruppamento e ordinamento.

Funzioni dei Report
Opzioni di Riepilogo

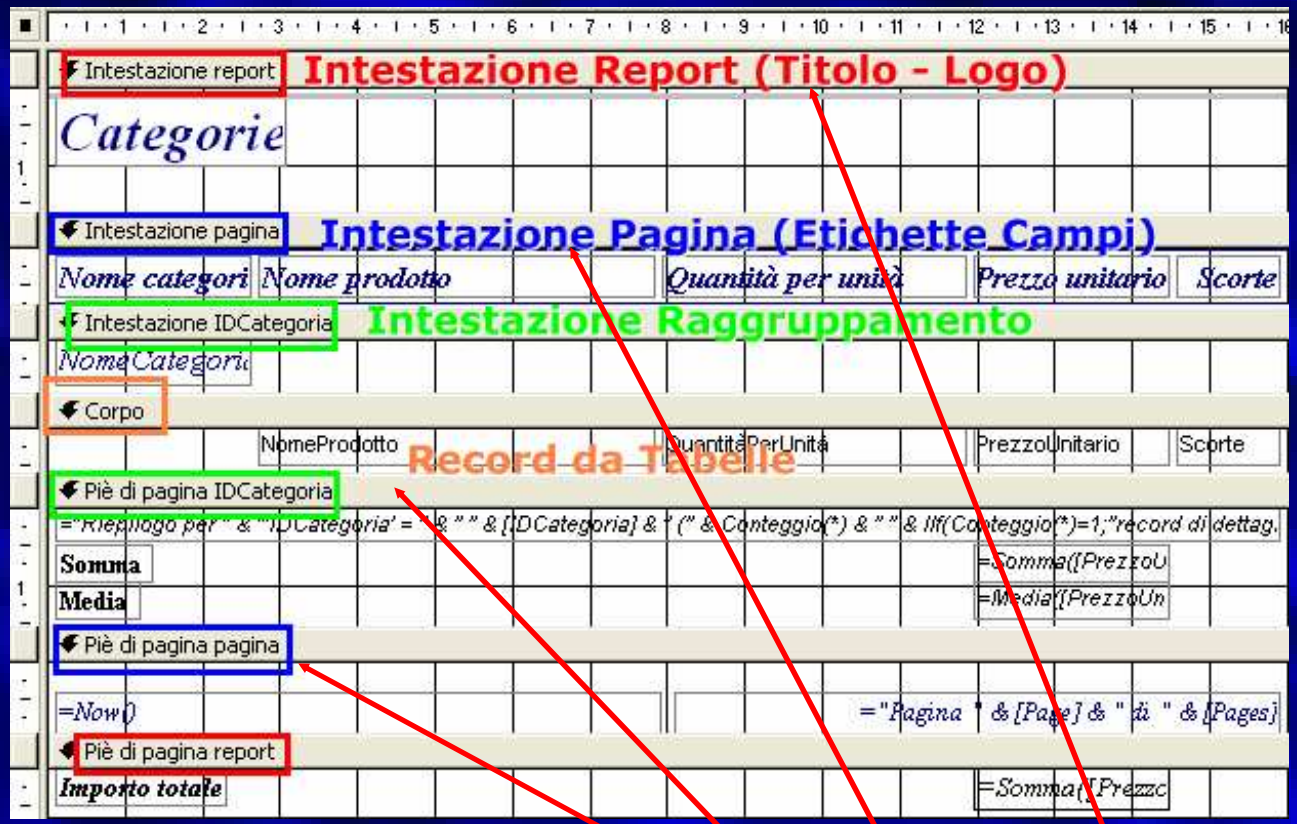


La funzione **Report** consente di riepilogare i dati

Effettuando ed inserendo in modo automatico alcuni calcoli per database che contengono campi numerici calcolabili.

Sono disponibili la Somma la Media, il valore Min e Max e la percentuale per le Somme.

Struttura di un Report



Sezioni di un Report

Il Generatore di Espressioni

E' uno strumento di appoggio utile alla creazione di espressioni da inserire nel database sfruttando operatori, funzioni egli oggetti del DB (tabelle, maschere, query e report)

Espressioni Comuni

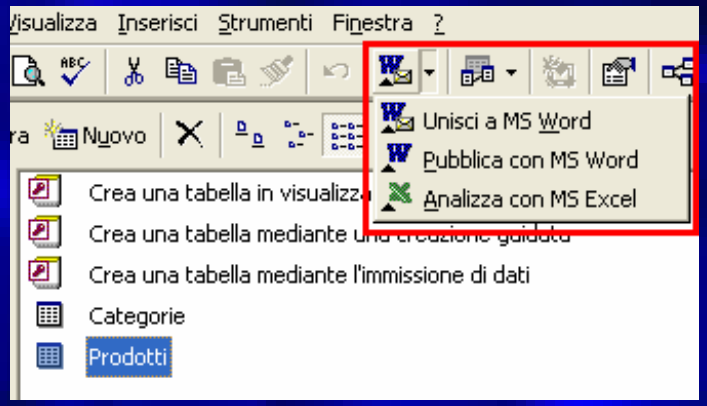
Espressione	Risultato
=[Page]	1, 2, 3
="Pagina " & [Page]	Pagina 1, Pagina 2, Pagina 3
="Pagina " & [Page] & " di " & [Pages]	Pagina 1 di 3, Pagina 2 di 3, Pagina 3 di 3
=[Page] & " di " & [Pages] & " Pagine"	1 di 3 Pagine, 2 di 3 Pagine, 3 di 3 Pagine
=[Page] & "/"& [Pages] & " Pagine"	1/3 Pagine, 2/3 Pagine, 3/3 Pagine
=[Country] & " - " & [Page]	UK - 1, UK - 2, UK - 3
=Format([Page], "000")	

Espressioni e numeri di pagina

Espressione	Risultato
=[Subtotale]+[Trasporto]	La somma dei valori dei campi Subtotale e Trasporto.
=[DataRichiesta]-[DataSpedizione]	La differenza tra i valori dei campi DataRichiesta e DataSpedizione.
=[Prezzo]*1,06	Il prodotto del valore del campo Prezzo e 1,06. Viene aggiunto il 6% al galore Prezzo.
=[Quantità]*[Prezzo]	Il prodotto dei valori dei campi Quantità e Prezzo.
=[TotaleImpiegato]/[TotalePaese]	Il quoziente dei valori dei campi TotaleImpiegato e TotalePaese.

Espressioni e Calcoli

Access97 verso Office



Unisci a MS Word: Consente di impostare la Stampa Unione in Word su un file nuovo e già esistente.

Pubblica con MS Word: crea un file di word (documento) con i record già intabellati

Analizza con Excel: importa la tabella selezionata direttamente in un foglio di Excel pronta per essere elaborata

Le Macro in Access

- ***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.

Approfondimento delle Macro

- Le Macro rappresentano il modo più semplice per scrivere un programma che automatizzi le funzioni del database. Sono costituite da un semplice elenco di istruzioni, denominate ***Azioni***, che vengono eseguite una di seguito all'altra, nell'ordine in cui il programmatore le ha scritte.
- La stesura di una macro è facilitata da un'apposita struttura a tabella che raccoglie tutti i comandi che sono stati inseriti nella macro stessa; all'interno di questa struttura è inoltre possibile affiancare ad *Azione* una breve descrizione per commentare la funzione svolta dalla *Azione* stessa.

i Moduli in Access (VBA)

- I Moduli rappresentano il modo più completo ed avanzato per scrivere un programma che automatizzi le funzioni del database
- un modulo è un insieme di dichiarazioni e routine (Function o Sub) di Visual Basic, Applications Edition memorizzate come una singola unità.
  Supportano il linguaggio Visual Basic per la costruzione delle Routine di Programmazione ed il linguaggio SQL per l'interrogazione e la manipolazione dei dati.

Per attivare una Routine da una Maschera per un controllo, invece di una Macro, impostare nelle *Proprietà del Controllo*, nella sezione *Evento*, alla voce *Su Clic:*

=NomeFunction ()

Moduli

- un modulo è un insieme di dichiarazioni e routine (Function o Sub) di Visual Basic. Di ROUTINE ce ne sono di due tipi:
  Sub -->Sub NomeRoutine ()
  Function -->Function NomeRoutine ()
  Ogni Routine viene chiusa da *End Sub* o *End Function* ripettivamente.
  All'interno di una Routine si trovano gli Argomenti ovvero nomi di variabili o costanti utilizzati dalle Routine stesse durante l'esecuzione.
- Una macro non gestisce gli errori, la sua interruzione non ne spiega il motivo
- I moduli in VBA consentono la gestione degli errori
- Esistono 2 tipi di moduli contenenti delle routine (o Function o Sub):
  - ***Moduli di classe***
  - ***Moduli standard***

Moduli di Classe

- Sono i *moduli di maschera e di report* associati ad una maschera o report contenenti delle routine evento che vengono eseguite in risposta ad un evento che si verifica sulla maschera o sul report (creati quando si crea la prima routine evento per una maschera o report).

Moduli Standard

- Contengono routine generali che non sono associate ad alcun altro oggetto e routine utilizzate di frequente che è possibile eseguire da un punto qualsiasi del database.

La Guida Rapida

- E' un ottimo strumento per avere informazioni rapide sulle funzionalità di Access.
- Dopo aver selezionato dal menu Help (?) la voce Guida Rapida, il cursore cambierà aspetto e basterà cliccare sulla voce interessata per avere spiegazione o generare l'apertura dell'Help in Linea.

Limiti di Access 97

- Gestisce al massimo file di 2 GB
- Non è dotato di un compilatore (ADK - access developer kit disponibile nelle versioni pro di office)
- Non gestisce il lock di riga
- Gestisce male gli accessi concomitanti al DB (max 15 utenti contemporaneamente)

Access 97 e L'Euro

- In tutte **le applicazioni Microsoft Office 97** è possibile immettere, visualizzare e stampare il simbolo dell'Euro, a condizione che nel sistema operativo del computer in uso siano presenti i caratteri e i driver necessari.
- Impostare la valuta desiderata sotto il Pannello di Controllo/Impostazioni internazionali
- **Maggiori informazioni:**
  - **http://www.microsoft.com/italy/euro/predisposizione/fase_3.asp**

Manualistica e Bibliografia

- http://www.microsoft.Com/access
- http://www.manualipc.It/manuali/corso/manuali.Php?Idcap=00&idman=17&size=&sid=
- http://www.netlink.it/manuali/