

Esercitazioni

Sistemi Informativi

Prof. Domenico Beneventano

**Università di Modena e Reggio Emilia
A.A. 2003/2004**

Ing. Andrea Bulgarelli

Ing. Stefania Magnani

Caratteristiche di MS Access

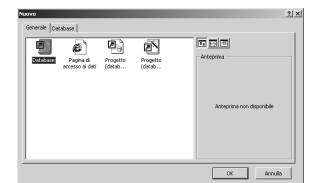
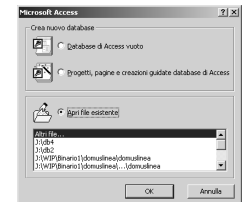
- Interazione a *finestre* o mediante SQL
- Funzioni (vista utente):
 - creazione e consultazione di tabelle, interfaccia tabellare
 - interfaccia a schede personalizzabili (un record per volta)
 - interfaccia SQL
 - salvataggio e modifica di interrogazioni
 - generazione di report
 - multiutenza
- Funzioni (vista sistema)
 - chiavi primarie e indici
 - valori nulli, regole di validità e domini di valori
 - gestione vincoli di integrità referenziale
 - programmazione

Database

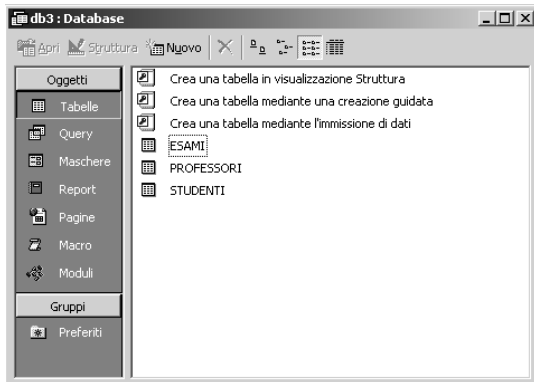
- è un *contenitore* di *tabelle, maschere, query, report, moduli, macro, pagine*
- le tabelle contengono i dati, gli altri tipi di documenti forniscono strumenti di accesso personalizzati
- ogni strumento di accesso può essere generato automaticamente in forma standard, poi modificato secondo le esigenze
- ogni strumento può essere *usato o modificato*
- è possibile importare/esportare DB da/verso i dbms più diffusi
- i dati possono essere facilmente collegati ad altri strumenti di ambiente windows (es: excel, word, ...)
- dal punto di vista DOS è un unico file con estensione .MDB

Creazione Database

- All'apertura di Access selezionare *Database di Access vuoto* oppure selezionare *Nuovo database* dal menu *File*
 - Nella finestra *Nuovo* selezionare l'icona *Database vuoto* nella scheda *Generale* e confermare con *OK*
 - Dalla finestra *Salva nuovo database* assegnare un nome al database da creare, scegliere la cartella in cui si desidera salvare il database e fare clic sul pulsante *Crea*.
- *Esempio: studente.mdb*



Finestra di database principale



Oggetti di un Database

- tipi di oggetti:
 - **tabelle**: contenitori di dati (relazionali)
 - **query**: operazioni su dati (di consultazione o modifica) e schema
 - **maschere**: interfacce visuali per l'accesso ai dati
 - **report**: produzione di stampe
 - **macro**: sequenze di operazioni pre-registrate e rieseguibili a comando
 - **moduli**: applicazioni programmate
 - **Pagine**: produzione di pagine Web per l'accesso ai dati
- ogni oggetto può essere creato, memorizzato, utilizzato, modificato
- tramite tabelle, maschere, query e pagine si possono anche modificare i dati

DDL

- CREATE TABLE e ALTER TABLE:
 - Proposizione CONSTRAINT
- CREATE INDEX
- CREATE PROCEDURE
- CREATE VIEW
- DROP TABLE, INDEX, PROCEDURE, VIEW

CREATE TABLE

- CREATE** [TEMPORARY] **TABLE** *tabella* (*campo1* *tipo* [(*dimensioni*)] [NOT NULL] [WITH COMPRESSION | WITH COMP] [*indice1*] [, *campo2* *tipo* [(*dimensioni*)] [NOT NULL] [*indice2*] [, ...]] [, CONSTRAINT *indicemulticampo* [, ...]])
- *tabella*: nome della tabella da creare.
 - *campo1*, *campo2* : nome del campo o dei campi da creare nella nuova tabella. È necessario creare almeno un campo.

CREATE TABLE (2)

- *tipo*: tipo di dati del *campo* nella nuova tabella.
- *dimensioni*: dimensioni del campo in caratteri, solo per campi con tipo di dati Testo e Binario.
- *indice1, indice2* : proposizione **CONSTRAINT** che definisce un indice a campo singolo.

Tipo di dati	Capacità di memorizzazione	Descrizione
BINARY	1 byte per carattere	In un campo di questo tipo è possibile memorizzare qualsiasi tipo di dati. Non viene effettuata alcuna conversione di dati (ad esempio, in testo). Il modo in cui l'input dei dati avviene in un campo binario determina il modo in cui viene visualizzato l'output dei dati.
BIT	1 byte	Valori Sì e No e i campi che contengono uno solo dei due valori.
TINYINT	1 byte	Un valore intero compreso tra 0 e 255.
MONEY	8 byte	Un numero intero proporzionato compreso tra - 922.337.203.685.477,5808 e 922.337.203.685.477,5807.
DATETIME (vedere DOUBLE)	8 byte	Un valore di data o di ora compreso tra 100 e 9999.
UNIQUEIDENTIFIER	128 bit	Un numero di identificazione univoco utilizzato con chiamate di routine remote.
REAL	4 byte	Un valore a virgola mobile a precisione singola con un intervallo compreso tra - 3,402823E38 e - 1,401298E-45 per i valori negativi e tra 1,401298E-45 e 3,402823E38 per i valori positivi e zero.

FLOAT	8 byte	Un valore a virgola mobile a precisione doppia con un intervallo compreso tra - 1,79769313486232E308 e - 4,94065645841247E-324 per i valori negativi e tra 4,94065645841247E-324 e 1,79769313486232E308 per i valori positivi e zero.
SMALLINT	2 byte	Un numero intero breve compreso tra -32.768 e 32.767 (vedere le note).
INTEGER	4 byte	Un numero intero lungo compreso tra - 2.147.483.648 e 2.147.483.647.
DECIMAL	17 byte	Tipo di dati numerico esatto con valori compresi tra 1028 - da 1 a - 1028 - 1. È possibile definire la precisione (da 1 a 28) e la proporzione (da 0 a precisione definita). La precisione e la proporzione predefinite sono rispettivamente 18 e 0.
TEXT	2 byte per carattere	Da zero ad un massimo di 2,14 GB.
IMAGE	Come necessario	Da zero ad un massimo di 2,14 GB. Utilizzato per oggetti OLE.
CHARACTER	2 byte per carattere	Da zero a 255 caratteri.

Proposizione CONSTRAINTS

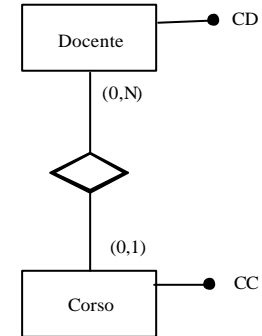
CONSTRAINT *nome*

{**PRIMARY KEY** (*primaria1* [, *primaria2* [, ...]]) |
UNIQUE (*univoca1* [, *univoca2* [, ...]]) |
NOT NULL (*nonnull1* [, *nonnull2* [, ...]]) |
FOREIGN KEY [NO INDEX] (*rif1* [, *rif2* [, ...]])
REFERENCES *tabellaesterna*
 [(*campoesterno1* [, *campoesterno2* [, ...]])]
 [ON UPDATE CASCADE | SET NULL]
 [ON DELETE CASCADE | SET NULL]}

CONSTRAINT (2)

- *nome*: nome del vincolo da creare.
- *primaria1, primaria2*: nome del campo o dei campi da definire come chiave primaria.
- *univoca1, univoca2*: nome del campo o dei campi da definire come chiave univoca.
- *nonnull1, nonnull2*: nome del campo o dei campi limitati a contenere valori non Null.
- *rif1, rif2*: nome di uno o più campi con funzione di chiave esterna che si riferiscono ai campi di un'altra tabella.
- *tabellaesterna*: nome della tabella esterna che contiene il campo o i campi specificati da *campoesterno*.
- *campoesterno1, campoesterno2*: nome del campo o dei campi in *tabellaesterna* specificati da *rif1, rif2*. È possibile omettere questa proposizione se il campo a cui si fa riferimento è la chiave primaria di *tabellaesterna*.

Esempio 1 : schema E/R



Esempio 1 : sc. relazionale

DOCENTE(CD, NOME, COGNOME,
CITTA)

CORSO(CC, NOME, CDOCENTE)
FK: CDOCENTE REFERENCES
DOCENTE

Esempio 1 : DDL

```
CREATE TABLE DOCENTE
(
  CD INTEGER,
  NOME text(30) NOT NULL,
  COGNOME text(30) NOT NULL,
  CITTA text(2),
  CONSTRAINT pk_docente PRIMARY KEY(CD)
);
```

Esempio 1 : DDL

```
CREATE TABLE CORSO
(
  CC char(2),
  Nome char(30) NOT NULL,
  CDOCENTE INTEGER,
  CONSTRAINT pk_corso PRIMARY KEY(CC),
  CONSTRAINT fk_corso_docente FOREIGN KEY
    (CDOCENTE) REFERENCES
    DOCENTE(CD)
);
```

Immissione dei dati

- Aprire la tabella in modalità **Foglio Dati** facendo doppio clic sul suo nome oppure tramite il pulsante Apri.
- Posizionarsi sull'ultimo record vuoto.
- Digitare il contenuto del record.

Successivamente è possibile:

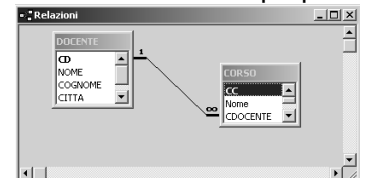
- Spostarsi all'interno della tabella tramite i tasti della tastiera o tramite mouse
- Selezionare e modificare il contenuto
- Eliminare un record
- Annullare le modifiche apportate
- Correggere gli errori di immissione

Personalizzazione Foglio Dati

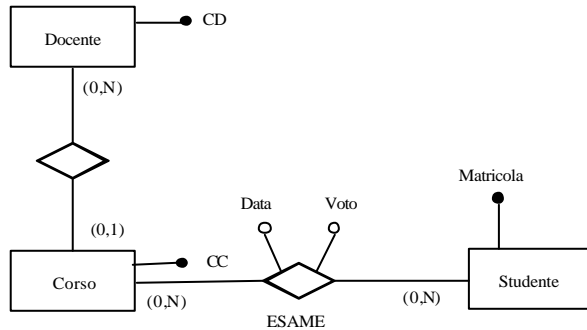
- E' possibile:
 - modificare l'ampiezza della colonna
 - bloccare le colonne in modo che restino sempre visualizzate
 - modificare l'altezza di una riga
 - visualizzare o nascondere le linee della griglia
 - modificare il tipo di carattere
 - salvare le modifiche di visualizzazione (Salva layout)

Esempio 1 : integrità

- Provare ad inserire dati e a comprendere:
 - Chiave primaria
 - Primo vincolo di integrità
 - Vincolo di integrità referenziale: le opzioni ON UPDATE CASCADE e ON DELETE CASCADE
- Utilizzare la *Visualizzazione foglio dati*
- Utilizzare la finestra *Strumenti\Relazioni* per la visualizzazione e la modifica delle proprietà delle relazioni



Estensione esempio 1



Estensione esempio 1 (2)

STUDENTE(Matricola, Nome, Cognome, Indirizzo, Citta, Note_studente)

ESAME(Matricola, CCorso, Data, Voto)

FK: Matricola REFERENCES STUDENTE

FK: CCorso REFERENCES CORSO

Estensione esempio 1 (3)

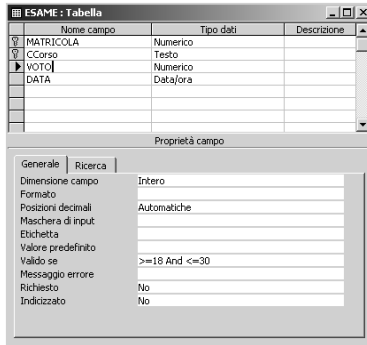
```
CREATE TABLE STUDENTE
(
  MATRICOLA INTEGER,
  NOME char(30) NOT NULL,
  COGNOME char(30) NOT NULL,
  INDIRIZZO char(40),
  CITTA char(2),
  NOTE_STUDENTE memo,
  CONSTRAINT pk_studenti PRIMARY
  KEY(MATRICOLA)
);
```

Estensione esempio 1 (4)

```
CREATE TABLE ESAME
(
  MATRICOLA INTEGER,
  CCorso CHAR(2),
  VOTO SMALLINT,
  DATA DATE NOT NULL,
  CONSTRAINT pk_esamisostenuti PRIMARY
  KEY(MATRICOLA,CCorso),
  CONSTRAINT fk_esami1 FOREIGN KEY (MATRICOLA)
  REFERENCES STUDENTE(MATRICOLA),
  CONSTRAINT fk_esami2 FOREIGN KEY (CCorso) REFERENCES
  CORSO
);
```

CHECK

- E' possibile impostare vincoli di check solamente da *Visualizzazione struttura*



ALTER TABLE

ALTER TABLE *tabella*

```
{  
ADD {COLUMN tipo campo[(dimensioni)] [NOT NULL]  
    [, CONSTRAINT ...] |  
ALTER COLUMN tipo campo[(dimensioni)] |  
    [CONSTRAINT ...] |  
DROP {COLUMN campo | CONSTRAINT  
    nomeconstraint}  
}
```

ALTER TABLE (2)

- Utilizzare ADD COLUMN per aggiungere un nuovo campo alla tabella.
- Utilizzare ALTER COLUMN per modificare il tipo di dati di un campo esistente.
- Utilizzare ADD CONSTRAINT per aggiungere un constraint.
- Utilizzare DROP COLUMN per eliminare un campo.
- Utilizzare DROP CONSTRAINT per eliminare un constraint.

ALTER TABLE (3)

- *tabella*: nome della tabella da modificare.
- *campo*: nome del campo da aggiungere, modificare o eliminare da *tabella*.
- *tipo*: tipo di dati di *campo*.
- *dimensioni*: dimensioni del campo in caratteri.
- *indice*: indice per *campo*.
- ...: vedi CONSTRAINT
- *nomeconstraint*: nome dell'indice multicampo da rimuovere.

ALTER TABLE - ESEMPIO

```
ALTER TABLE CORSO
ADD COLUMN DATA_INIZIO DATE,
      DATA_FINE DATE, ORARIO CHAR(50),
      DURATA_CORSO INTEGER
```

- Provare ad aggiungere l'anno accademico alla tabella corso
- Aggiungere i campi indirizzo, telefono, e-mail, e data di nascita alla tabella docente

ADD COLUMN E NOT NULL

- Provare ad eseguire il seguente statement su una tabella DOCENTE che contiene già dei record:

```
ALTER TABLE DOCENTE
ADD COLUMN CF CHAR(16) NOT
NULL
```

- In Access funziona! In MS SQL Server e Sybase ASA non funziona.
- Il comportamento di Access non è corretto perché **il database si trova ora in uno stato inconsistente!**

ALTER TABLE – ESEMPIO (2)

- Esempio di modifica del tipo di dato di una colonna:

```
ALTER TABLE CORSO
ALTER COLUMN ORARIO MEMO
```

- Esempio di eliminazione di una colonna:

```
ALTER TABLE CORSO
DROP DURATA_CORSO
```

ALTER TABLE (4)

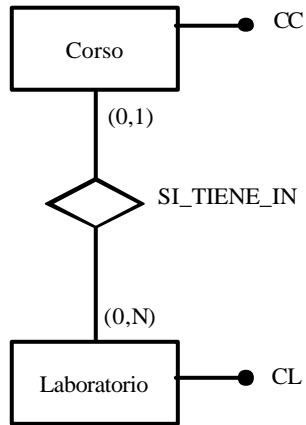
- In MS SQL Server e Sybase ASA la sintassi può essere diversa:

```
ALTER TABLE CORSO
ADD data_inizio date,
ADD orario char(50);
```
- In Sybase ASA è inoltre possibile anche rinominare colonne:

```
ALTER TABLE CORSO
RENAME orario TO orario_lezione;
```
- Inoltre, in Access non è possibile combinare comandi di tipo diverso:

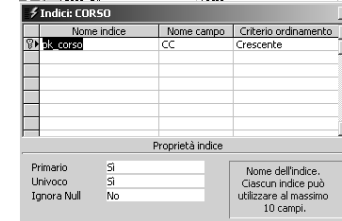
```
ALTER TABLE CORSO
ADD data_fine date,
DROP durata_corso;
```

Esercizio ALTER TABLE



CREATE INDEX

- Da SQL:
CREATE [UNIQUE] INDEX indice
ON tabella (campo [ASC|DESC]), campo
[ASC|DESC], ...)
[WITH { PRIMARY | DISALLOW NULL | IGNORE
NULL }]
- In *Visualizzazione struttura*, menu *VisualizzaIndici*



CREATE INDEX (2)

- UNIQUE: impedisce la presenza di valori duplicati.
- Nella proposizione opzionale WITH, è possibile applicare regole di convalida dei dati. È possibile:
 - Impedire voci Null nel campo o nei campi indicizzati di nuovi record utilizzando l'opzione DISALLOW NULL.
 - Impedire che i record con valori Null nel campo o nei campi indicizzati vengano inclusi nell'indice utilizzando l'opzione IGNORE NULL.
 - Definire il campo o i campi indicizzati come chiave primaria utilizzando la parola riservata PRIMARY. Ciò implica che la chiave sia univoca, in modo che si possa omettere la parola riservata UNIQUE.

DROP

- DROP TABLE nome tabella
- DROP INDEX nome indice ON nome tabella
- DROP PROCEDURE nome procedura
- DROP VIEW nome view

Esercizio DDL 1

- Creare un database con il DBMS Access con le caratteristiche descritte nello schema relazionale seguente. Generare le tabelle necessarie ed indicare le associazioni esistenti fra di esse gestendo i vincoli di integrità referenziale secondo le seguenti specifiche:
 - Se una camera viene eliminata, eliminare tutti i servizi relativi a quella camera.
 - Impedire l'eliminazione di un cliente se esistono delle prenotazioni da lui effettuate.

Esercizio DDL 1 (2)

CAMERA(numero, categoria, numeroLetti, piano, costo, appartamento, numeroCamera)
CHECK (categoria = 'Normale' OR categoria = 'Lussuosa')
CHECK (numeroLetti <=4)

PRENOTAZIONE(numeroCamera, arrivo partenza, codCliente)
F.K. (numeroCamera) REFERENCES CAMERA(numero)
F.K. (codCliente) REFERENCES CLIENTE(codCliente)
codCliente NOT NULL

CLIENTE(codCliente nome, cognome, nazionalità, età)

SERVIZIO(codServizio, descrizione)

SERVIZIO_EXTRA(codServizio, costoExtra)
F.K. (codServizio) REFERENCES SERVIZIO(codServizio)

SERVIZI_CAMERA(numeroCamera, codServizio)
F.K. (numeroCamera) REFERENCES CAMERA(numero)
F.K. (codServizio) REFERENCES SERVIZIO(codServizio)

Esercizio DDL 2

- Creare un database con il DBMS Access con le caratteristiche descritte nello schema relazionale seguente. Generare le tabelle necessarie ed indicare le associazioni esistenti fra di esse gestendo i vincoli di integrità referenziale secondo le seguenti specifiche:
 - Se un servizio viene eliminato eliminare in cascata tutti i servizi di viaggi ad esso relativi.
 - Impedire l'eliminazione di un cliente se esistono delle prenotazioni da lui effettuate.

Esercizio DDL 2 (2)

VIAGGIO(numero, descrizione, categoria, durata, costo)
CHECK (categoria = 'Pacchetto' OR categoria = 'Crociera')
CHECK (durata <= 21)

PRENOTAZIONE(numeroViaggio, codCliente, data)
F.K. (numeroViaggio) REFERENCES VIAGGIO(numero)
F.K. (codCliente) REFERENCES CLIENTE(codCliente)
codCliente NOT NULL

CLIENTE(codCliente nome, cognome, nazionalità, età)

SERVIZIO(codServizio, descrizione, servizioExtra, costoExtra)

SERVIZI_VIAGGIO(numeroViaggio, codServizio)
F.K. (numeroViaggio) REFERENCES VIAGGIO(numero)
F.K. (codServizio) REFERENCES SERVIZIO(codServizio)

SELECT

```
SELECT [ DISTINCT | ALL ] <lista-select>  
FROM <lista-from>  
[WHERE <condizione>]  
[ORDER BY <lista-order>]
```

SELECT (2)

- <lista-select>:
 - Uno o più attributi specificati delle tabelle di <lista-from>
 - Funzioni aggregate: COUNT, AVG, MIN, MAX, SUM
 - *
 - Costanti
- <lista-from>
 - Tabelle, viste o join
- <lista-order>
 - Uno o più attributi di <lista-select>
 - ASC | DESC

SELECT (3)

```
SELECT [ DISTINCT | ALL ] <lista-select>  
FROM <lista-from>  
[WHERE <condizione>]  
[GROUP BY <lista-group>]  
[HAVING <condizione>]  
[ORDER BY <lista-order>]
```

SELECT (4)

- <lista-select>
 - Uno o più attributi di <lista-select>
 - Funzioni di aggregazione

Esempi select

- Tutti i clienti con più di 30 anni
- Tutti i clienti francesi
- Tutti i viaggi di tipo 'Pacchetto' con costo inferiore ai €1000
- Tutti i viaggi di tipo 'Pacchetto' a Roma
- Combinare le due query precedenti

- Provare altri esempi in modo libero...

ESEMPIO PROVA

- Con riferimento a *Esercizio DDL 2*, provare le seguenti interrogazioni

ESEMPIO PROVA (2)

- Riportare per ogni viaggio il suo numero ed il totale di prenotazioni effettuate (non interessa il caso dei viaggi senza prenotazioni)

```
SELECT NUMEROVIAGGIO, COUNT(*)  
FROM PRENOTAZIONE  
GROUP BY NUMEROVIAGGIO
```

ESEMPIO PROVA (3)

- Riportare per ogni categoria di viaggio il numero totale di prenotazioni effettuate (non interessa il caso di categorie di viaggi senza prenotazioni)

```
SELECT V.categoria, Count(*)  
FROM VIAGGIO AS V, PRENOTAZIONE AS P  
WHERE V.numero=P.NUMEROVIAGGIO  
GROUP BY V.categoria;
```

ESEMPIO PROVA (4)

- Selezionare i viaggi di categoria 'Crociera' con più di 10 prenotazioni

```
SELECT V.NUMERO
FROM VIAGGIO AS V, PRENOTAZIONE AS P
WHERE V.NUMERO = P.NUMEROVIAGGIO
AND V.CATEGORIA = 'crociera'
GROUP BY V.NUMERO
HAVING COUNT(*) > 10
```

ESEMPIO PROVA (5)

- Selezionare il codice dei clienti che hanno fatto più di 5 prenotazioni di viaggi di categoria 'Crociera'.

```
SELECT CODCLIENTE, COUNT(*)
FROM VIAGGIO AS V, PRENOTAZIONE AS P
WHERE V.NUMERO = P.NUMEROVIAGGIO
AND CATEGORIA = 'CROCIERA'
GROUP BY CODCLIENTE
HAVING COUNT(*) > 5
```

ESEMPIO PROVA (6)

- Selezionare i viaggi di costo superiore a 1000 che sono stati prenotati (esiste almeno una prenotazione) tra il 15/09/03 e il 30/03/04.

```
SELECT V.NUMERO, V.DESCRIZIONE, V.CATEGORIA, V.DURATA, V.COSTO
FROM VIAGGIO AS V
WHERE V.NUMERO In (SELECT P.NUMEROVIAGGIO
FROM PRENOTAZIONE AS P
WHERE P.DATA BETWEEN #15/09/2003#
AND #30/03/2004#)
AND V.COSTO > 1000 ;
```

INSERT

```
INSERT INTO <tabella> [(<lista-attr>)]
VALUES (<lista-valori>)
```

```
INSERT INTO S(matr, Snome, Città,
Acorso) VALUES('M1', 'Andrea', 'MN',
2)
```

INSERT(2)

```
INSERT INTO <tabella> [(<lista-attr>)]  
<subquery>
```

```
INSERT INTO MEDIA  
SELECT Matr, AVG(voto)  
FROM E  
GROUP BY Matr
```

UPDATE

```
UPDATE <tabella> SET <attr> =  
<espressione>  
[WHERE <condizione>]
```

```
UPDATE E  
SET Voto = 1.1*Voto  
WHERE Voto < 24
```

UPDATE (2)

- Aumentare il prezzo di tutti i viaggi di categoria 'Crociera' del 20%.

```
UPDATE VIAGGIO  
SET VIAGGIO.COSTO = 1.2* VIAGGIO.COSTO  
WHERE VIAGGIO.CATEGORIA= "crociera";
```

DELETE

```
DELETE FROM <tabella>  
[WHERE <condizione>]
```

- Cancellare tutti i clienti che non hanno fatto alcuna prenotazione

```
DELETE FROM CLIENTE  
WHERE CODCLIENTE IN (  
SELECT CODCLIENTE  
FROM CLIENTE AS C  
WHERE CODCLIENTE NOT IN (SELECT DISTINCT CODCLIENTE  
FROM PRENOTAZIONE  
)
```

SECONDO ESEMPIO PROVA

COMUNE(COD_COM, NOME, PR, N_ABITANTI)

AZIENDA(PIVA, RAG_SOC, INDIRIZZO, COM)
F.K. COM REFERENCES COMUNE (COD_COM)

PRODOTTO(COD_PROD, DESCRIZIONE,
ANNO_LANCIO)

LISTINO(PIVA, COD_PROD, PREZZO, SCONTO)
F.K. PIVA REFERENCES AZIENDA (PIVA)
F.K. COD_PROD REFERENCES PRODOTTO
(COD_PROD)

SECONDO ESEMPIO PROVA (2)

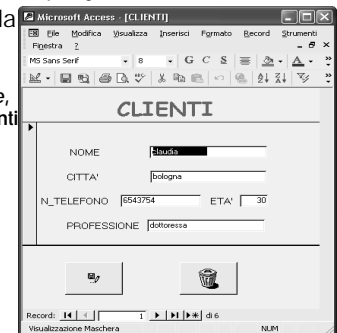
- Creare le tabelle **utilizzando il linguaggio SQL**
- Gestire i vincoli di integrità referenziale con le seguenti specifiche
 - Impedire la cancellazione di un prodotto (dalla tabella PRODOTTO) se tale prodotto è presente in listino
 - Se una azienda viene cancellata, si devono cancellare tutti i suoi elementi collegati in LISTINO
 - Se una azienda cambia partita IVA, si deve riportare la modifica anche su tutti i suoi elementi collegati in LISTINO

SECONDO ESEMPIO PROVA (3)

- Scrivere in SQL le seguenti interrogazioni
 - Elencare tutte le aziende di 'Reggio Emilia' (cioè del comune con nome 'Reggio Emilia')
 - Elencare tutte le aziende della provincia di 'Reggio Emilia' (cioè di un comune con provincia 'Reggio Emilia')
 - Elencare il codice_prodotto ed il prezzo dei prodotti delle aziende di 'Modena'
 - Eliminare dal listino prezzi tutti i prodotti delle aziende di 'Modena'
 - Riportare per ogni azienda la PIVA ed il numero totale di prodotti che sono in LISTINO
 - Riportare la PIVA delle aziende con più di 50 prodotti in LISTINO

Maschere

- maschera di visualizzazione, aggiornamento e inserimento
- può essere generata automaticamente a partire dalla struttura di una tabella o di una query
- può essere personalizzata da un progettista che funzionale
- **menu a tendina, pulsanti di opzione, procedure da associare ai vari eventi (pressione del mouse in un punto determinato, rilascio del mouse, ...)**
- **procedure di controllo per vincoli di integrità**



Maschere

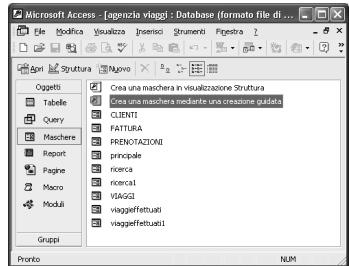
- consentono di visualizzare e gestire in modo diverso i dati contenuti nelle tabelle

Creazione di una maschera:

- Visualizzare la scheda *Maschere*

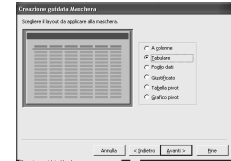
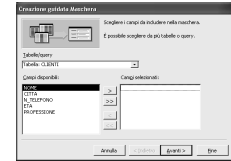
- Si può utilizzare:

- funzione di *autocomposizione*
- modalità *Visualizzazione Struttura*



Autocomposizione Maschera

- Si seleziona il nome della tabella o della query
- Si selezionano i campi della tabella
- Selezionare il formato di visualizzazione:
 - a colonne
 - tabulare
 - foglio dati
- Specificare lo stile di visualizzazione
- Specificare il titolo

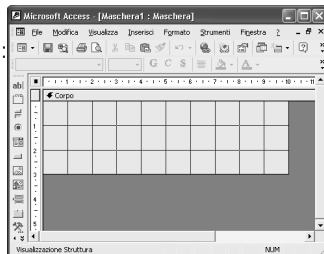


Creazione maschera vuota

- scegliere modalità visualizzazione struttura
- è possibile aggiungere:
 - caselle di testo
 - etichette
 - altri controlli (pulsanti, casella di controllo, casella combinata etc...)

La finestra di lavoro è costituita da:

- area di lavoro
- strumenti
 - barra degli strumenti
 - righe
 - casella degli strumenti



Sezioni di una maschera

- **Una maschera può essere suddivisa in cinque sezioni :**
 - Intestazione maschera
 - Intestazione pagina
 - Corpo
 - Piè di pagina pagina
 - Piè di pagina maschera
- di default viene visualizzato solo il corpo
- è possibile modificare le dimensioni di ciascuna sezione e specificare le proprietà

Maschere basate su più tabelle

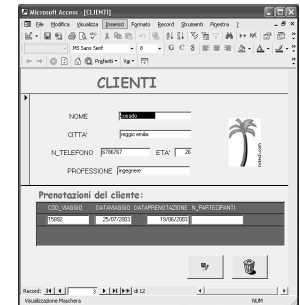
Esistono **tre** modalità:

- basare la maschere su una query
- selezionare campi di diverse tabelle tramite la procedura di autocomposizione tabella
- creare una sottomaschera che consente di visualizzare la relazione esistente tra i record di diverse tabelle

Sottomaschere

→ vengono usate soprattutto per visualizzare i record contenuti in tabelle collegate con relazioni del tipo uno a molti

- è possibile crearle con la modalità autocomposizione:
1. creare la maschera principale
 2. utilizzare lo strumento Sottomaschera/Sottoreport per creare la sottomaschera sulla griglia della maschera principale

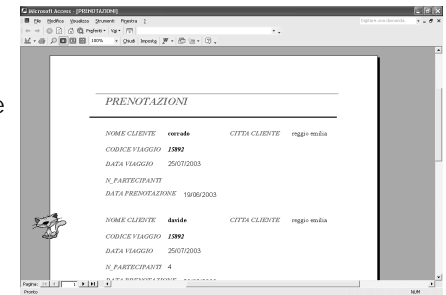


Creazione di una sottomaschera

- Creare la maschera principale
- Creare la sottomaschera
- Riaprire la maschera principale in visualizzazione struttura
- Trascinare dalla finestra di database il nome della sottomaschera sulla griglia della maschera principale
- Verificare che nella tavola delle proprietà della sottomaschera le proprietà *Collega campi secondari* e *Collega campi master* contengano i campi che collegano la maschera principale e la sottomaschera

Report

- per la produzione di stampe
- personalizzare le intestazioni
- definire gruppi e sottogruppi
- calcolare totali parziali sui gruppi



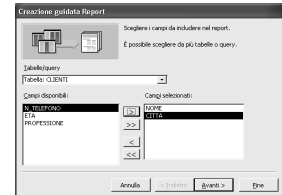
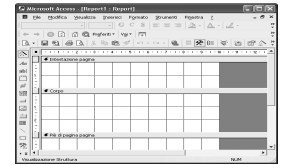
Report

- Un report è una presentazione di dati, provenienti da una o più tabelle, formattata e stampata.
- Può essere basato su una tabella o su una query
- Può essere utilizzato per una stampa dei dati, ma non per la loro immissione o modifica

Creazione di un report

Esistono tre modalità:

- Creazione tramite la funzione *Report Standard*
 - *formato a colonne*
 - *formato tabulare*
- Creazione tramite autocomposizione
 - selezione dei campi
 - livello di raggruppamento dati
 - tipo di ordinamento
 - layout e stile
- Creazione report vuoto



Finestra di struttura di un report

- È simile a quella di una maschera
- È suddiviso in sezioni:
 - Intestazione report
 - Intestazione pagina
 - Intestazione nome_raggruppamento
 - Corpo
 - Piè di pagina nome_raggruppamento
 - Piè di pagina pagina
 - Piè di pagina report

Ordinamento e raggruppamento dei dati

Si può impostare:

- Tramite la procedura di autocomposizione selezionando il campo secondo il quale si desidera raggruppare i record
- Direttamente nella struttura del report
 - fare clic sullo strumento *Ordinamento* e raggruppamento della barra degli strumenti
 - selezionare il campo su cui basare l'ordinamento o il raggruppamento
 - selezionare il criterio di ordinamento
 - se si vuole usare il campo per il raggruppamento e avere una intestazione di gruppo modificare in Sì la proprietà *Intestazione (gruppo)*
 - se si vuole usare il campo per il raggruppamento e avere un piè di pagina di gruppo, modificare in Sì la proprietà *Piè pagina (gruppo)*

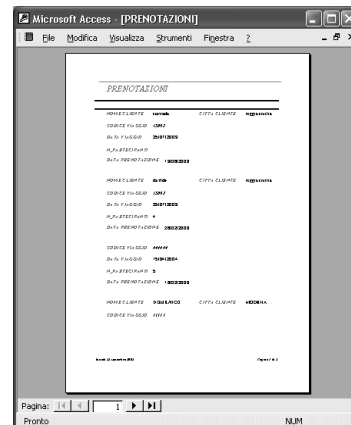
Calcoli in un report

- Dalla finestra si struttura inserire una casella di testo standard non associata
- Per immettere un'espressione:
 - digitare l'espressione nella casella iniziando con il segno uguale (=), oppure
 - aprire la tavola delle proprietà e digitare l'espressione nella proprietà *Origine controllo* o utilizzare il generatore di espressioni
- se si colloca il campo calcolato in una sezione relativa ad un gruppo il calcolo si riferisce al gruppo, se lo si colloca in una sezione relativa al report il calcolo si riferisce a tutti i record, se lo si colloca nella sezione Corpo, il calcolo si basa solo sul record

Aggiunta di numeri di pagina e date

Si possono utilizzare:

- le funzioni =Date() o =Now() =Pagina in un controllo calcolato;
- utilizzare i comandi del menu *Inserisci*
 - Numeri di pagina...
 - Data e ora...



Macro

- Una macro è un'istruzione che consente di eseguire automaticamente una o più operazioni (azioni)
- Le macro possono essere usate:
 - ricerca dati
 - verifica dei dati immessi
 - impostare proprietà di oggetti
 - utilizzare più maschere contemporaneamente
 - collegare maschere con report
 - trasferire dati
 - applicazioni personalizzate

Creazione di una macro

- Dalla finestra macro selezionare il pulsante nuovo. La sezione *Argomento azione* consente di impostare gli argomenti delle azioni immesse nella colonna *Azione*



Le azioni

- Le azioni sono istruzioni che indicano quali operazioni compiere
- Possono essere aggiunte in due modi
 - trascinando il nome dell'oggetto dalla finestra di database in una cella della colonna Azione
 - fare clic su una cella della colonna Azione e selezionare un'azione dall'elenco visualizzato
- Nel primo caso gli argomenti vengono impostati automaticamente, nel secondo caso è necessario impostare gli argomenti dell'azione

Esecuzione di una macro

- E' possibile eseguire una macro:
 - dalla finestra macro tramite lo strumento *Esegui*
 - dalla finestra di database facendo doppio clic sul nome della macro o usando il bottone *Esegui*
 - da un'altra finestra selezionando *Macro...* dal menu *Strumenti* e specificando il nome della macro
 - all'interno di un'altra macro tramite l'azione *EseguiMacro*
 - in conseguenza di un evento che si verifica in una maschera o in un report

Macro condizionate

- Consente di eseguire una macro solo se si verifica una determinata condizione
- Per aggiungere una condizione ad una macro:
 - aprire la macro in modalità struttura
 - visualizzare la colonna *Condizione* facendo clic sullo strumento corrispondente
 - digitare nella colonna *Condizione* in corrispondenza dell'azione che si desidera condizionare, l'espressione logica che determina la condizione
 - se si desidera porre un'unica condizione per diverse azioni, si immettano dei puntini di

Risposta ad eventi di una maschera

- E' possibile creare uno o più macro, le quali rispondono a particolari eventi che si possono verificare durante l'utilizzo di una maschera
- Aprire la maschera in visualizzazione struttura
- Visualizzare le tavole delle proprietà
- Fare clic sulla proprietà relativa all'evento al quale si desidera che la macro risponda e selezionare il nome della macro oppure generare una nuova macro tramite il pulsante *Genera*

Risposta agli eventi relativi ai singoli controlli

- Una macro può essere collegata ai singoli controlli di una maschera.
- Aprire la maschera in visualizzazione struttura
- Selezionare il controllo
- Aprire la tavola delle proprietà
- Fare clic sulla proprietà relativa all'evento e selezionare o generare la macro

Debug di un macro

- E' possibile controllare ogni singola azione per verificare i risultati di ciascuna (modalità *passo a passo*)
- Aprire la finestra *Macro* corrispondente alla macro da controllare
- Fare clic sullo strumento *Passo a passo*
- Eseguire la macro utilizzando le normali procedure
- In tal modo viene visualizzata la finestra di dialogo *Macro passo a passo* che contiene la descrizione dell'azione da eseguire e tre bottoni: