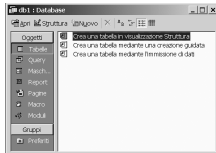


# Basi di dati

## Il linguaggio SQL



teoria ... e pratica con Microsoft Access

### Riepilogando ...

- Nelle basi di dati esiste
  1. una parte **invariante** nel tempo , lo **schema**, costituita dalle caratteristiche dei dati (nomi degli attributi, domini, ...)
  2. una parte **variabile**, detta **istanza**, costituita dai valori effettivi

2

### Riepilogando ...

- In Access abbiamo visto come definire lo schema di una base di dati usando il pulsante



... che permette la definizione delle tabelle

- E abbiamo anche visto come **popolare** la base di dati per creare l'istanza usando il pulsante



... che permette di "riempire le righe" delle tabelle

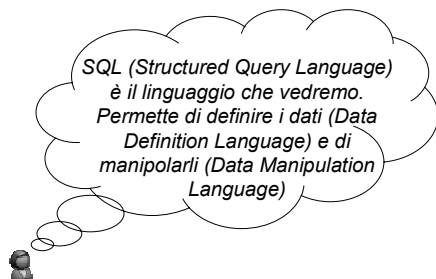
3

### Riepilogando ...

- A questo punto siamo in grado di rappresentare il "mondo" che vogliamo modellare e di raccogliere le informazioni di interesse
- Ma questo non basta ... vogliamo anche poter modificare le informazioni e, soprattutto, estrarre dalla totalità dei dati delle informazioni specifiche
- ... in altre parole, abbiamo bisogno di un linguaggio per l'**interrogazione** e la **manipolazione** dei dati ...

4

## SQL



5

### Quali operazioni?

- **Proiezione**  
estrazione di attributi (colonne) da una tabella
- **Selezione**  
estrazione di tuple (righe) che verificano un certo criterio da una tabella
- **Join**  
estrazione di informazioni correlate, contenute in tabelle diverse

6

## Esempio di proiezione

T1			
Nome	Cognome	Nato il	Nato a
Anna	Rossi	2/2/71	TO
Gigi	Bianchi	23/4/80	Ivrea
Iris	Bianchi	15/9/45	CN

La **proiezione** di T1 sugli attributi Nome e Cognome restituisce

T2	
Nome	Cognome
Anna	Rossi
Gigi	Bianchi
Iris	Bianchi

7

## Esempio di selezione

T1			
Nome	Cognome	Nato il	Nato a
Anna	Rossi	2/2/71	TO
Gigi	Bianchi	23/4/80	Ivrea
Iris	Bianchi	15/9/45	CN

La **selezione** dei record di T1 tali che "Nato il >= 1/1/1960" restituisce

T2			
Nome	Cognome	Nato il	Nato a
Anna	Rossi	2/2/71	TO
Gigi	Bianchi	23/4/80	Ivrea

8

## Esempio di join

T1			T2	
Titolo	Autore	Codice	Utente	Cod libro
Poesie	Rossi	111	Pippo	111
Prosa	Verdi	222	Pippo	222
Elegie	Verdi	333	Pluto	111

Il **join** fra le due tabelle restituisce

T3				
Titolo	Autore	Codice	Utente	Cod libro
Poesie	Rossi	111	Pippo	111
Poesie	Rossi	111	Pluto	111
Prosa	Verdi	222	Pippo	222

9

## Casi estremi

Nome	Reparto	Tel	Reparto
Ugo	segreteria	111111	centralino
Oreste	produzione	222222	direzione

join vuoto

Nome	Reparto	Tel	Reparto
Ugo	segreteria	111111	segreteria
Oreste	segreteria	222222	segreteria

prodotto cartesiano

10

## Natural join

Correla i dati sulla base di valori uguali su attributi con lo stesso nome ed evita le ripetizioni

Titolo	Autore	Codice	Nome	Codice
Poesie	Rossi	111	Bianchi	111
Poesie	Rossi	111	Neri	111
Prosa	Verdi	222	Bianchi	222

Ripetizione!

Titolo	Autore	Codice	Nome
Poesie	Rossi	111	Bianchi
Poesie	Rossi	111	Neri
Prosa	Verdi	222	Bianchi

Esistono altre varianti dell'operatore di join che non vedremo

11

## Il linguaggio SQL

Creato negli anni '70 presso IBM

Inizialmente era un linguaggio di interrogazione

Ora è diventato il linguaggio di riferimento per le basi di dati relazionali

Standardizzato grazie al lavoro di ISO (International Standard Organization) e ANSI (American National Standard Institute)

12

## Il linguaggio SQL

Fornisce le funzionalità per la definizione dello schema di una base di dati (Data Definition Language)

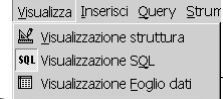
Fornisce le funzionalità per la modifica e l'interrogazione dell'istanza di una base di dati (Data Manipulation Language)

Vedremo in linguaggio mediante esempi

13

## Access e il linguaggio SQL

Access (come vedremo) permette di operare sui dati in modalità grafica, ovvero tramite i pulsanti e i menu della sua interfaccia. Tuttavia, ad ogni operazione sui dati corrisponde del codice SQL che viene generato in modo automatico. Si può vedere questo codice mediante il menu **Visualizza**



14

## Il linguaggio SQL

Permette di **creare** nuove tabelle, **modificare** le tabelle esistenti, **inserire**, **aggiornare**, **cancellare** i dati dalle tabelle


Una tabella è una collezione ordinata di **attributi**, ciascuno con il proprio **dominio**, e con eventuali **vincoli**

15

## Creazione di una tabella

```
create table nome_tabella  
(  
    descrizione della tabella  
)
```

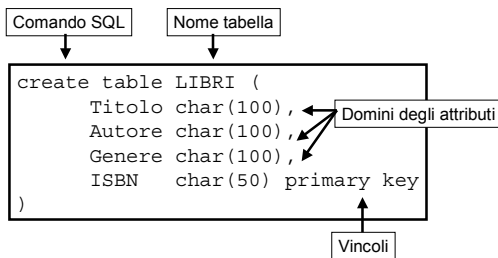
Per ogni attributo si deve definire un **nome**, un **dominio** ed eventualmente un insieme di **vincoli**

 Ricordate i passi di Access per la definizione della struttura di una tabella?

16

## Creazione di una tabella: esempio

LIBRI (Titolo, Autore, Genere, ISBN)




17

## Domini elementari

Indicano i valori che un attributo può assumere e le operazioni che possono essere compiute su di esso

per le stringhe di caratteri	character (n)
	varchar (n)
per i numeri interi	integer
per i numeri con virgola	float
per le date	date

 Ricordate i domini di Access?

18

## Vincoli intrarelazionali

Sono associati agli attributi

default valore	indica il valore che un attributo deve avere quando viene inserito un nuovo record in cui non viene specificato alcun valore per quell'attributo
not null	il valore nullo non è ammesso per quell'attributo
unique	il valore può comparire una volta sola
primary key	chiave primaria

☑ Ricordate i vincoli di Access?

19

## Esempio

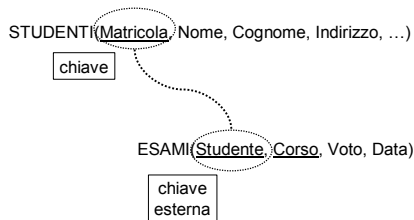
```
create table LIBRI
(
    Titolo      varchar(200) not null,
    Autore      varchar(200) default 'anonimo',
    ISBN        varchar(13)  not null,
    Collocazione varchar(10) unique,
    primary key (ISBN)
)
```

```
create table PRODOTTI
(
    Codbarre     varchar(15) not null,
    Marca        varchar(100),
    Costo        float default 0,
    Dataacquisto date,
    Scadenza     date,
    primary key (Codbarre)
)
```

20

## Vincoli interrelazionali

Sono sostanzialmente vincoli di **integrità referenziale**, definiti tramite la definizione di **chiavi esterne (foreign key)**



21

## Vincoli interrelazionali

In SQL si utilizza la parola chiave **references** oppure **foreign key**

PRODOTTI (Codbarre, Marca, Costo, ... , ...)

```
create table PRODOTTI (
    Codbarre  varchar(15) not null,
    Marca     varchar(100),
    Costo     float default 0,
    Dataacquisto date,
    Scadenza  date,
    primary key (Codbarre)
)
```

22

## Vincoli interrelazionali

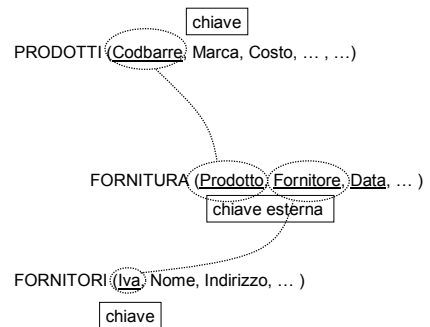
In SQL si utilizza la parola chiave **references** oppure **foreign key**

FORNITURA (Prodotto, Fornitore, Data, ... )

```
create table PRODOTTI (
    create table FORNITURA (
        Prodotto varchar(15)
            references PRODOTTI (Codbarre),
        Fornitore varchar(18)
            references FORNITORI (Iva),
        Data date,
        primary key (Prodotto, Fornitore, Data)
    )
)
```

23

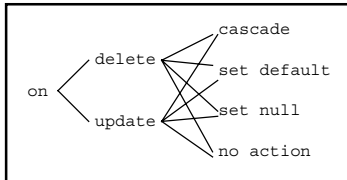
## Vincoli interrelazionali



24

## Violazioni ed azioni

È possibile specificare come reagire alle violazioni dei vincoli di integrità referenziale

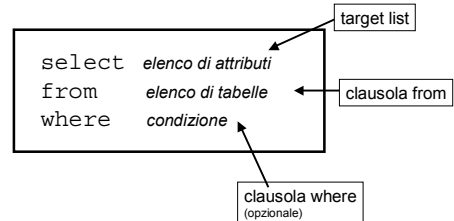


Ricordate i vincoli di integrità referenziale di Access?

25

## Interrogazione: select

SQL permette di specificare **cosa** si vuole ottenere e non come ottenerlo (linguaggio dichiarativo)



26

## Esempio

PERSONALE			
Nome	Cognome	Qualifica	Tel
Andrea	Rossi	dirigente	338
Renzo	Verdi	segretario	331
Orestina	Verdi	segretario	331
Giulia	Bianchi	tecnico	332
Giulio	Neri	tecnico	332
Renzo	Neri	consulente	332

```
select Cognome, Qualifica from PERSONALE
```

Cognome	Qualifica
Rossi	dirigente
Verdi	segretario
Verdi	segretario
Bianchi	tecnico
Neri	tecnico
Neri	consulente

(proiezione)

27

## Esempio

PERSONALE			
Nome	Cognome	Qualifica	Tel
Andrea	Rossi	dirigente	338
Renzo	Verdi	segretario	331
Orestina	Verdi	segretario	331
Giulia	Bianchi	tecnico	332
Giulio	Neri	tecnico	332
Renzo	Neri	consulente	332

```
select * from PERSONALE where Tel=331
```

Nome	Cognome	Qualifica	Tel
Renzo	Verdi	segretario	331
Orestina	Verdi	segretario	331

(selezione)

28

## Esempio

OPERE	
Titolo	Autore
Poesie	Bianchi
Poesie	Rossi
Storia	Rossi
Italiano	Verdi

AUTORI		
Cognome	Nascita	Residenza
Bianchi	.....	Verona
Rossi	.....	Padova
Verdi	.....	Genova

```
select OPERE.Titolo, OPERE.Autore, AUTORI.nascita
from OPERE,AUTORI
where ( OPERE.Autore = AUTORI.Cognome and
       AUTORI.Residenza = 'Padova' )
```

Titolo	Autore	Nascita
Poesie	Rossi	...
Storia	Rossi	...

29

## Clausola where

La clausola where ammette come argomento una espressione booleana costruita combinando semplici predicati con gli operatori and, or, not

Ogni predicato usa gli operatori =, <, >, <=, >=, <> per confrontare i valori degli attributi con delle espressioni

Nel caso più semplice si confronta il valore di un solo attributo, ma ci possono essere casi più complessi

30

## Clausola where: esempio

```
select * from IMPIEGATO where  
(Dipartimento = 'amministrazione' or  
Dipartimento = 'produzione' )
```

```
select * from DISCHI where  
(Genere = 'rock' and Durata > 60 )
```

```
select * from PERSONE where  
(Nascita < #1/1/1942# and Sesso='F')
```

31

## Interrogazione: join

AUTORI ( Cognome, Nome, Nascita, ...)

OPERE ( Titolo, Autore, Anno, Genere, ...)

```
select *  
from AUTORI join OPERE  
on (AUTORI.Cognome=OPERE.Autore)
```

tabelle

attributi da correlare

Ne ripareremo in Access

32

## Funzioni di aggregazione

A volte si desidera estrarre informazioni **elaborando** i dati

calcolare il numero medio di libri letti da ciascun utente della biblioteca

calcolare il profitto massimo relativo alle vendite nell'anno 1998

calcolare l'età dei partecipanti a un corso in base all'anno di nascita

calcolare i prezzi + IVA dei prodotti su di un catalogo ...

Esistono delle funzioni che si possono usare, per es. min, max, somma, conteggio, media ... le vedremo in Access

33

## Altre operazioni

Select e join **non** consentono di modificare il contenuto della base di dati

Azioni desiderate

Inserimento di nuovi dati

Aggiornamento di dati

Cancellazione di dati

34

## Inserimento di nuovi dati

```
insert into nometabella  
( elenco attributi )  
values  
( elenco valori )
```

Attributi e valori vanno separati da una virgola

Per i valori ci sono delle convenzioni: le stringhe di testo vanno scritte tra apici mentre le date vanno scritte tra # ... #

35

## Inserimento di nuovi dati: esempio

```
insert into PERSONALE  
(Nome, Cognome, Qualifica, Nascita)  
values  
( 'Marco', 'Pautasso', 'tecnico', #3/7/1975#)
```

```
insert into UTENTI  
(Cognome, Tel, Codice)  
values  
( 'Vassallo', 453334, 'U004' )
```

36

## Inserimento di nuovi dati: esempio

UTENTI

Cognome	Indirizzo	Tel	Email	Codice
---------	-----------	-----	-------	--------

```
insert into UTENTI
(Cognome, Tel, Codice)
values
('Vassallo', 453334, 'U004')
```

Cognome	Indirizzo	Tel	Email	Codice
Vassallo R.	null	453334	null	U004

In corrispondenza dei campi non indicati verrà inserito valore null

37

## Modifica di dati

```
update nometabella set
...
attributoi = expr
...
where condizione
```



Se non specificate nessuna condizione le modifiche verranno effettuate su tutta la tabella!

38

## Modifica di dati: esempio

PRODOTTI

Nome	Costo	Genere
torcetti	2	dolce
canestrelli	2.5	dolce
focaccia	0.9	salato
margherita	2	salato
Meringhe	1.8	dolce

```
update PRODOTTI set
Costo = Costo * 1.2
where (Genere='dolce')
```

In un colpo solo si incrementa il costo di tutti i dolci di un certo valore percentuale

39

## Cancellazione di dati

Cancellazione di una tabella

```
drop table nometabella
```

```
drop table PRODOTTI
```

Cancellazione di record

```
delete from nometabella where
condizione
```

```
delete from PRODOTTI
where (Costo > 1.9)
```



Se non specificate nessuna condizione si cancellano tutti i record della tabella!

40

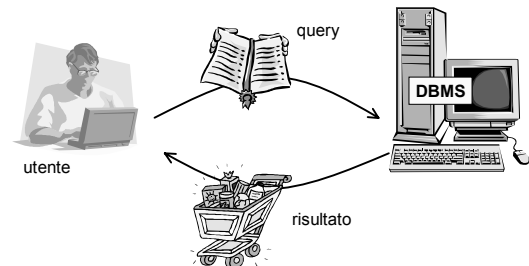
## SQL

Anche per SQL ci sono altri comandi più complicati che non vedremo. Descrizioni dettagliate si trovano sui libri di basi da dati. Ad esempio potete consultare il cap. 4 del libro *Basi di Dati di Atzeni, Ceri, Paraboschi, Torlone*, casa editrice McGraw-Hill



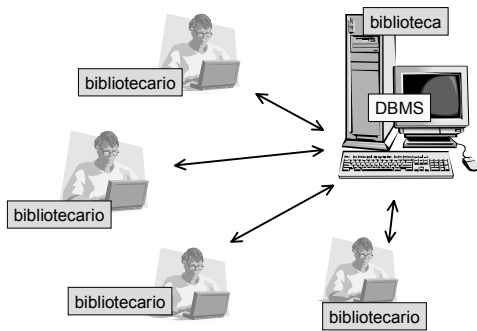
41

## DBMS – architettura client/server



42

## DBMS – molti utenti!



43

## DBMS – molti utenti!

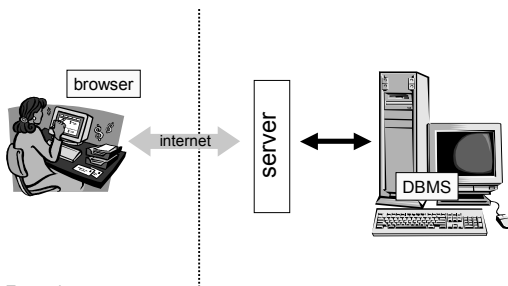
- I vari utenti non si accorgono gli uni degli altri. Ciascuno si interfaccia con il DBMS come se fosse il **solo** utente del sistema
- Le elaborazioni sono eseguite dal processore e in molti sistemi il processore è unico!
- Il DBMS gestisce le varie interazioni eseguendo un **parallelismo virtuale**, ovvero eseguendo alcune istruzioni relative all'interazione con il primo utente, alcune di interazione con il secondo e così via



Non entreremo nei dettagli di come viene gestito l'accesso concorrente ai dati ... ma questo parallelismo virtuale non vi ricorda nulla?

44

## Basi di dati e il web



Esempi  
banche on line, negozi virtuali, motori di ricerca, banche dati aziendali, riviste on-line ...

45

## Basi di dati e il web

- l'utente dichiara la propria identità (login e password)
- l'utente esegue delle operazioni
  - estrazione di informazioni (es. contenuto del conto corrente)
  - modifica di dati (es. emissione di un bonifico)
- l'utente lavora in parallelo con molti altri utenti
- i dati dell'utente sono riservati e devono essere persistenti
- l'interfaccia è costituita da un browser web
- i dati sono conservati in una base di dati

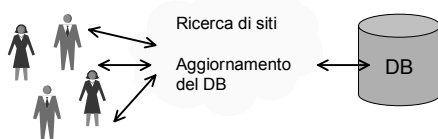
46

## Esempio: motori di ricerca

Motori di Ricerca

Altavista  
Yahoo  
Google  
Ask.com  
Lycos  
Virgilio  
...

Gestiscono informazioni sui siti web e sono accessibili via web



47

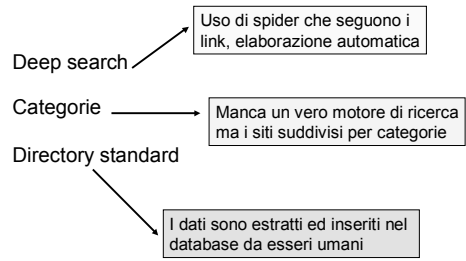
## Esempio: interrogazione



48



## Esempio: aggiornamento



49