

DBMS

Un DBMS deve garantire:

- **Affidabilità**
- **Privatezza dei dati**
- **Efficienza**
- **Efficacia**

DBMS

Affidabilità

Un DBMS deve garantire di poter mantenere intatto il suo contenuto, anche in caso di malfunzionamento.

L'integrità dei dati è affidata a procedure di **backup** (salvataggio) e **recovery** (recupero) dei dati, o alla loro **duplicazione** nei casi più critici.

DBMS

Privatezza dei dati

Ogni utente, abilitato a utilizzare la base di dati attraverso una procedura di riconoscimento, può accedere ad insiemi limitati di dati e compiere solo certe operazioni su di essi.

DBMS

Efficienza

Un DBMS deve operare e fornire risposte agli utenti in tempi accettabili, utilizzando una quantità il più possibile limitata di risorse.

L'efficienza dipende essenzialmente dalle tecniche utilizzate per l'implementazione del DBMS e dalla buona progettazione della base di dati.

Si misura (come in tutti i sistemi informatici) in termini di *tempo di esecuzione* (tempo di risposta) e *spazio di memoria* (principale e secondaria) occupato.

DBMS

Efficacia

Capacità di un DBMS di rendere produttive le attività degli utenti, cioè di consentire la realizzazione di basi di dati che risolvano in modo efficace i problemi degli utenti.

Concetto generico, qualitativo e non legato a specifiche funzionalità del DBMS. Non esistono criteri oggettivi per valutarla.

Modelli dei dati

Un modello di dati è costituito dai *concetti* sulla base dei quali i dati sono strutturati e codificati. Ogni modello di dati fornisce meccanismi di strutturazione dati, analoghi ai costruttori di tipo dei linguaggi di programmazione.

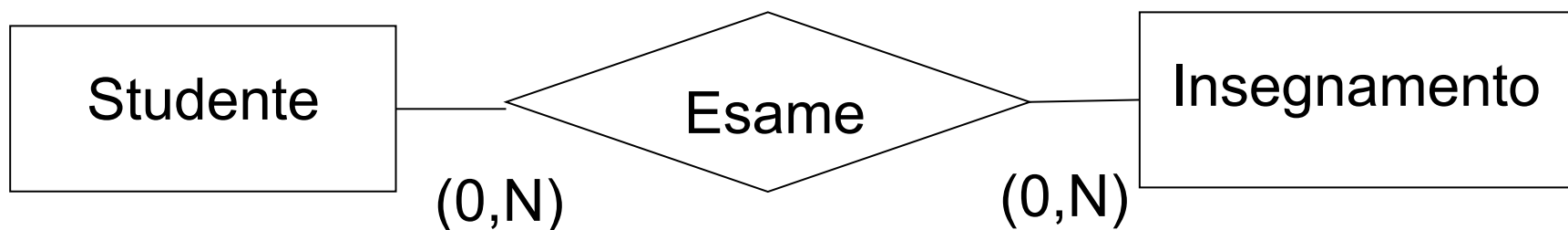
I ***modelli concettuali*** descrivono la realtà mediante concetti astratti, ma soggetti a precise regole. Non sono finalizzati alla rappresentazione dei dati, ma ai concetti del mondo reale di cui i dati sono *istanze*.

Si usano in fase di ***progettazione***.

Nel corso useremo i *diagrammi Entità/Relazione*

Modelli dei dati

*Esempio (molto semplice) di diagramma Entità/Relazione, modello di tipo **concettuale**.*



Si esprimono i concetti che devono essere rappresentati dai dati come concetti indipendenti (**entità**) e relazioni logiche che li collegano (**relazioni**) e che esistono solo in presenza delle entità.

Modelli dei dati

I DBMS si differenziano in base al *modello logico* che utilizzano. I **modelli logici**, seppure astratti, riflettono la *struttura* con cui i dati sono organizzati.

- **Relazionale**

Il più diffuso, basato su un modello tabellare dei dati

- **Gerarchico**

Utilizzato nei primi DBMS (anni 60), tuttora utilizzato, basato su strutture ad albero

- **Reticolare**

Estensione del modello gerarchico, basato su grafi

- **A oggetti**

Estensione del modello relazionale basato sui paradigmi OOP.

- **XML**

(semistrutturato) Deriva dal modello gerarchico, ma è più flessibile

Modello Relazionale

E' un *modello logico*: definisce tipi attraverso il costruttore *relazione*, che organizza i dati secondo **record a struttura fissa**, rappresentabili attraverso tabelle.

Es. (relazioni INSEGNAMENTO e MANIFESTO)

Corso	Titolare
Basi di Dati e Web	Cagnoni
Fondamenti di Informatica	Tomaiuolo
Ingegneria Del Software	Poggi

CdL	Materia	Anno
IET-I	Basi di Dati e Web	3
IET	Fondamenti di Informatica	1
IET-I	Ingegneria del Software	3

Schemi

In ogni base di dati si possono distinguere:

- Lo **schema**, sostanzialmente invariante nel tempo. Ne descrive la struttura (*aspetto intensionale*)
Nell'esempio, le tabelle e le «intestazioni» delle tabelle
- Le **istanze**, cioè i valori attuali, che possono cambiare anche molto rapidamente
(*aspetto estensionale*)
Nell'esempio, i singoli dati contenuti in ciascuna tabella



Schemi

Lo *schema* di una base di dati è la parte *dichiarativa* ed *invariante* della base di dati e ne definisce la struttura.

Nel modello relazionale lo schema di una relazione è paragonabile alla definizione del prototipo di una funzione in C.

INSEGNAMENTO (Corso, Titolare)

è lo *schema* della relazione INSEGNAMENTO definita su due *attributi* (campi del record identificati da una etichetta che ne descrive il significato).

Le n-ple di *attributi* appartenenti alla relazione, a ciascuno dei quali deve essere assegnato un valore, sono dette *istanze* della relazione.

(Basi di dati, Cagnoni)

è una istanza di INSEGNAMENTO.

Schemi

Gli schemi possono operare a diversi livelli di astrazione

- **Schema interno (o schema fisico)**
implementa lo schema logico mediante strutture fisiche di memorizzazione
- **Schema logico**
descrive la struttura dell'intera base di dati mediante il modello logico adottato dal DBMS (reticolare, gerarchico, relazionale)

Schemi

- **Schema esterno**

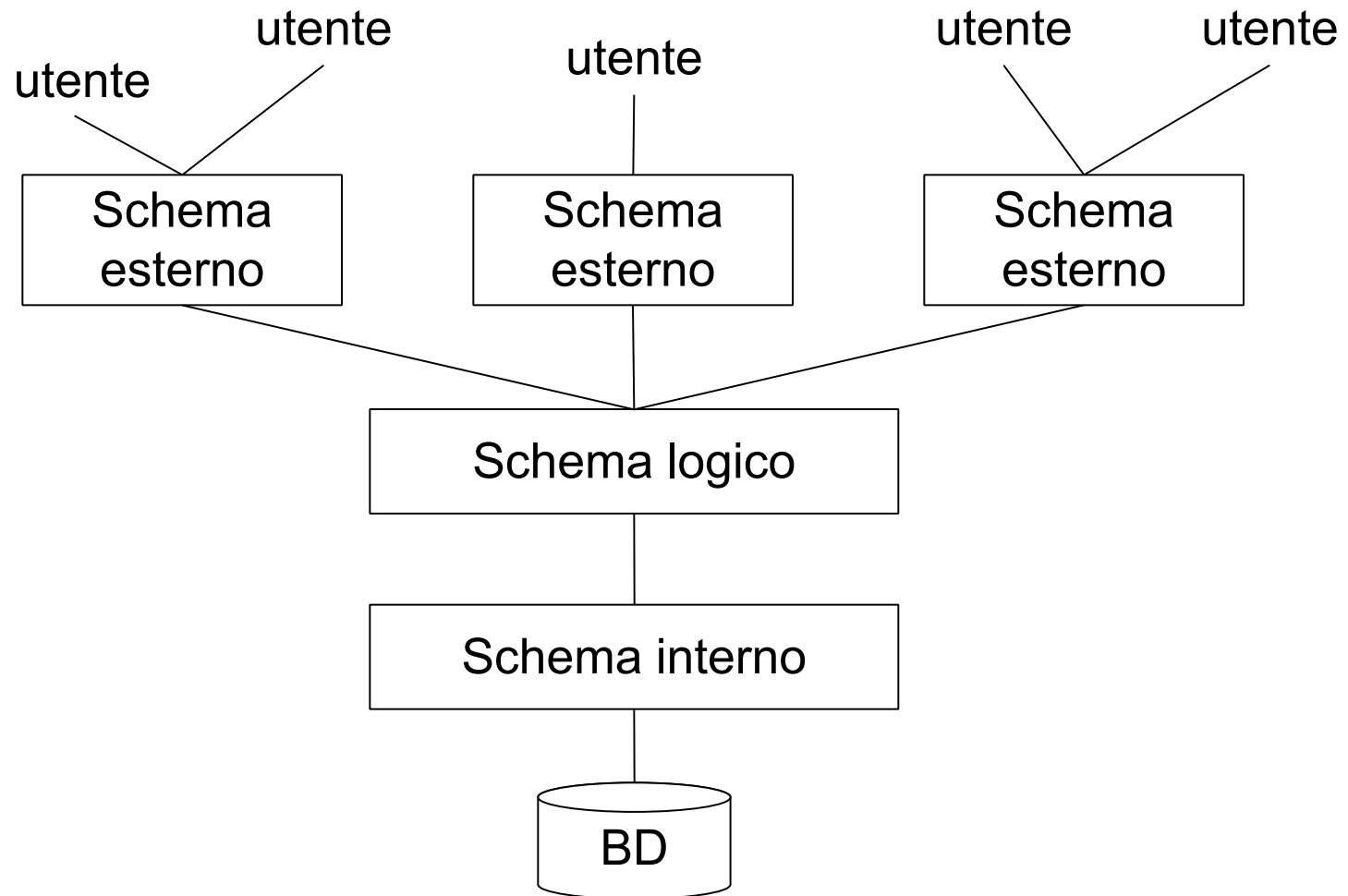
descrive la struttura di una porzione della base di dati attraverso il modello logico, dal punto di vista di una classe di utenti. Generalmente realizzato mediante *viste*, relazioni derivate da quelle che costituiscono lo schema logico.

Es. (i soli corsi del curriculum Ingegneria Informatica)



CdL	Materia	Anno
IET-I	Basi di Dati e Web	3
IET-I	Ingegneria del Software	3

Architettura standard (ANSI/SPARC) a tre livelli per DBMS



Progettazione di una base di dati

I diversi tipi di modello riflettono le diverse fasi della progettazione: date le specifiche del problema da risolvere

- **Progettazione concettuale**

(specifiche → schema concettuale)

Definisce quali concetti rappresentare, sia dal punto di vista della loro descrizione che da quello delle relazioni logiche che esistono fra di essi

- **Progettazione logica**

(schema concettuale → schema logico)

Definisce come rappresentare i concetti introdotti a livello concettuale in forma di strutture dati

- **Progettazione fisica** (schema logico → schema fisico)

Definisce come allocare e gestire fisicamente i dati all'interno del calcolatore

Accesso alla base di dati

- Linguaggi testuali interattivi
- Comandi inclusi in estensioni di linguaggi tradizionali
- Comandi inclusi in linguaggi di sviluppo ad hoc
- Interfacce grafiche amichevoli

Linguaggi per basi di dati

- **Linguaggi di definizione dei dati**

Utilizzati per definire gli schemi e le autorizzazioni per l'accesso

- **Linguaggi di manipolazione dei dati**

Utilizzati per l'interrogazione e l'aggiornamento dei contenuti della base di dati

Alcuni linguaggi specializzati (es. SQL) presentano le caratteristiche di entrambi i tipi di linguaggio.

SQL, un linguaggio interattivo

```
SELECT Corso, Aula, Piano  
FROM Aule, Corsi  
WHERE Nome = Aula  
AND Piano = "Terra"
```



Corsi

Corso	Aula
Sistemi	N3
Reti	N2

Nome	Piano
N3	Terra
N2	Terra

Aule

Corso	Aula	Piano
Sistemi	N3	Terra
Reti	N2	Terra

SQL immerso in linguaggio ad alto livello

```
write('nome della citta"?'); readln(citta);  
EXEC SQL DECLARE P CURSOR FOR   
    SELECT NOME, REDDITO  
    FROM PERSONE  
    WHERE CITTA = :citta ;  
EXEC SQL OPEN P ;  
EXEC SQL FETCH P INTO :nome, :reddito ;  
while SQLCODE = 0 do begin  
    write('nome della persona:', nome, 'aumento?');  
    readln(aumento);  
    EXEC SQL UPDATE PERSONE SET REDDITO = REDDITO +  
        :aumento  
        WHERE CURRENT OF P  
    EXEC SQL FETCH P INTO :nome, :reddito  
end;  
EXEC SQL CLOSE CURSOR P
```

PHP embedded in codice HTML

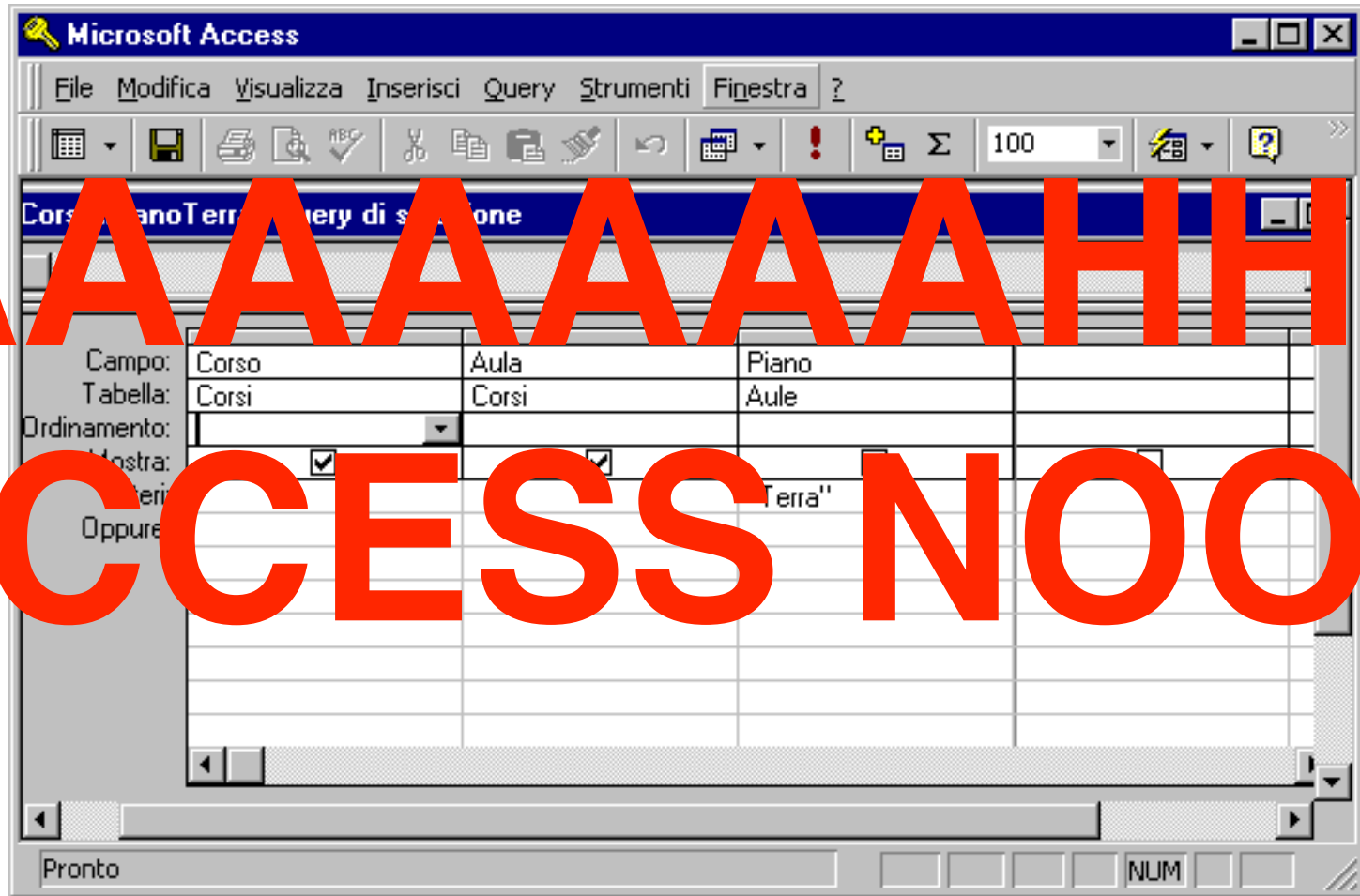
```
<html> <body>
<?php
$user="root"; $password=""; $host="127.0.0.1"; $database="prova";
mysql_connect($host,$user,$password);
@mysql_select_db($database) or die("Unable to select database");
$query="SELECT * from utenti";
$res=mysql_query($query);
mysql_close();

$row = mysql_fetch_assoc($res);
print '<table><tr>';
foreach($row as $name => $value) {
print "<th>$name</th>";}
print '</tr>';

while($row) {
print '<tr>';
foreach($row as $value) {
print "<td>$value</td>";}
print '</tr>';
$row = mysql_fetch_assoc($res);}
print '</table>';
?> </body> </html>
```

Connessione a mySQL
tramite API PHP

Interazione non testuale (in Access)



Realizza l'approccio detto "Query by Example"