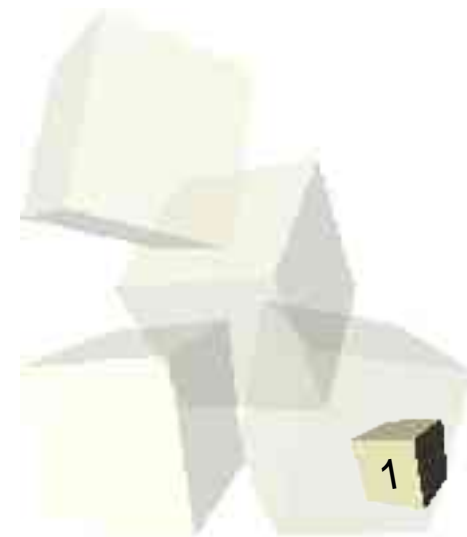


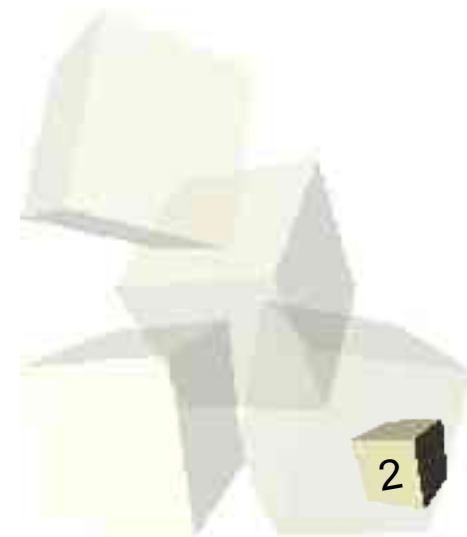
# Le basi di dati

dalla teoria all'utilizzo di tutti i giorni


Alessandro Tanasi  
alessandro@lonerunners.net  
<http://www.lonerunners.net>



- Il problema: **organizzare dati in modo comodo e efficiente**
- La risposta: **il sistema informativo**  
procedure e automatismi che gestiscono le informazioni necessarie all'azienda, di solito con mezzi informatici



## ■ Database



Una collezione di dati, che contengono le informazioni di interesse per il nostro sistema informativo

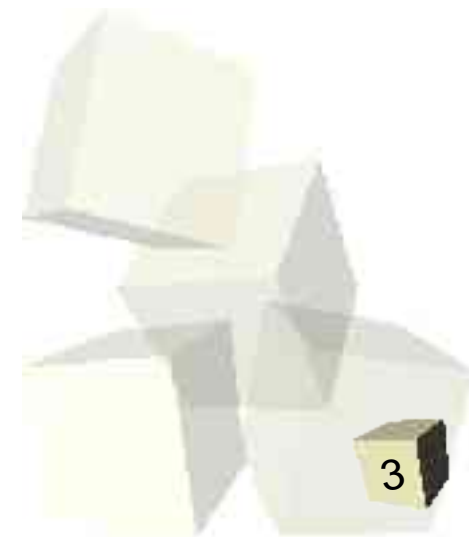
## ■ Gestore del Database

(DBMS Data Base Management System)



Si occupa della corretta gestione dei nostri dati (database)

- ♦ integrità
- ♦ sicurezza
- ♦ affidabilità
- ♦ efficienza
- ♦ condivisione



I dati sono memorizzati in un **modello relazionale**[1] ideato da Codd[2] e riassunto nelle sue 12 regole.

Relazione:

- ogni riga (record) è diversa dalle altre
- i dati possono essere associati

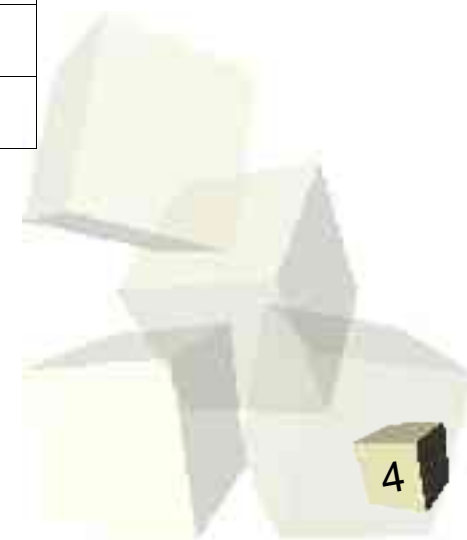
Si opera su tabelle

Studente

<u>Matricola</u>	Nome	Cognome
1	Pippo	Rossi
2	Pluto	Verdi

[1] <http://www.acm.org/classics/nov95/toc.html>

[2] [http://en.wikipedia.org/wiki/Edgar\\_F.\\_Codd](http://en.wikipedia.org/wiki/Edgar_F._Codd)



## Creare strutture complesse con le tabelle

### Studente

<b>Matricola</b>	<b>Nome</b>	<b>Cognome</b>
1	Pippo	Rossi
2	Pluto	Verdi

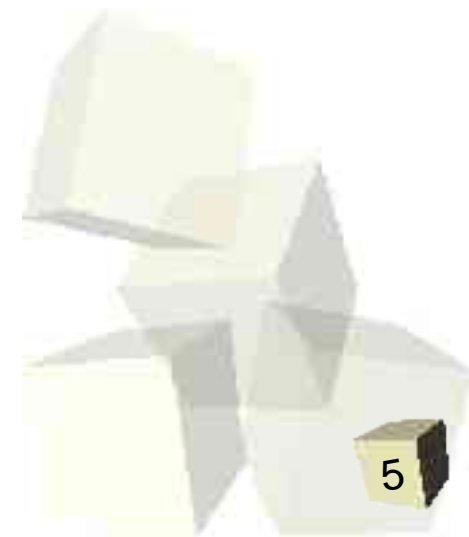
### Esami

<b>Matricola</b>	<b>Materia</b>
1	Informatica
2	Economia
2	Matematica

Studenti ed esami possono essere associati, l'operazione si chiama join.

### Esami fatti da un studente

<b>Nome</b>	<b>Cognome</b>	<b>Materia</b>
Pippo	Rossi	Informatica
Pluto	Verdi	Economia
Pluto	Verdi	Matematica



Molteplicità o cardinalità delle relazioni:

→ **uno a uno**: un record di una tabella è in relazione con un'unica riga dell'altra

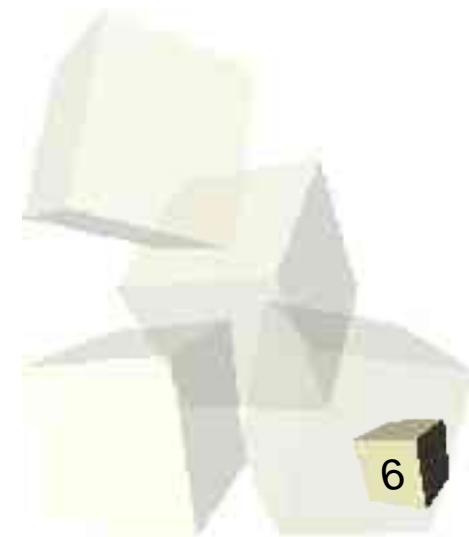
*es. una persona ha una residenza*

→ **uno a molti**: un record di una tabella è in relazione con più righe dell'altra

*es. uno studente ha fatto molti esami*

→ **molti a molti**: vari record di una tabella sono in relazione con varie righe dell'altra

*es. un libro ha molti autori*



La chiave permette di garantire univocità ai record, è quindi obbligatoria nella maggior parte dei casi.

### Studente

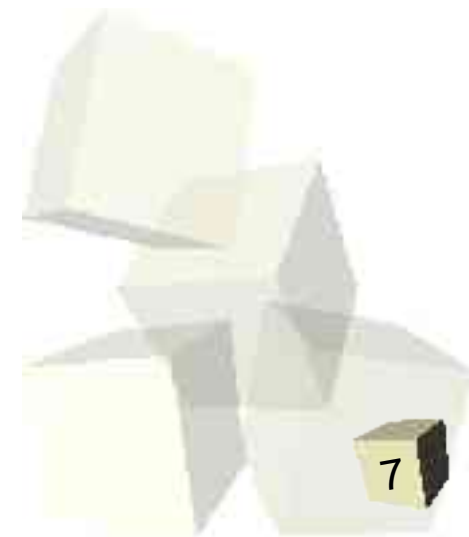
<u>Matricola</u>	Nome	Cognome
1	Pippo	Rossi
2	Pluto	Verdi

### Esami

<u>Matricola</u>	Materia
1	Informatica
2	Economia
2	Matematica

La **chiave primaria** identifica il record, si indica sottolineando il campo.

- ♦ valore diverso per ogni record
- ♦ non può essere nulla
- ♦ di solito è un numero progressivo
- ♦ può essere composta da più campi



L'univocità permette di eseguire associazioni, si può avere un riferimento a una chiave di un'altra tabella

Studente

<u>Matricola</u>	Nome	Cognome
1	Pippo	Rossi
2	Pluto	Verdi

Docente

<u>Matricola</u>	Nome	Studente
doc1	Paperino	2
doc2	Topolino	1

chiave esterna



La **chiave esterna** è un campo in una tabella che corrisponde alla chiave primaria di un'altra. Servono per la creazione di relazioni.



## Dove sono gli errori?

Studente

<u>Matricola</u>	<u>Nome</u>	<u>Cognome</u>
1	Pippo	Rossi
2	Pluto	Verdi
3	Gigi	

Esami

<u>Matricola</u>	<u>Voto</u>	<u>Nome</u>
1	39	Informatica
2	24	Economia
9	19	Economia
2	15 e Lode	Matematica

I **vincoli di integrità** sono regole che ci impediscono di inserire dati errati, e mantengono la base di dati consistente.

Un DBMS viene istruito per impedire la creazione di situazioni inconsistenti.



Esempio: Pluto incassa un assegno fatto da Paperino

Le operazioni coinvolte sono:



- ✓ togliere l'importo dal conto di Paperino
- ✓ aggiungere l'importo al conto di Pluto

Queste operazioni devono essere eseguite **entrambe correttamente**

Le **transazioni** ci permettono di tentare l'esecuzione di un set di istruzioni, e nel caso di un errore annullano le operazioni eseguite.



# SQL Structured Query Language

Il linguaggio utilizzato per “parlare” con il DBMS e per rappresentare i dati.

**Query** : richiesta, interrogazione che poniamo al DBMS

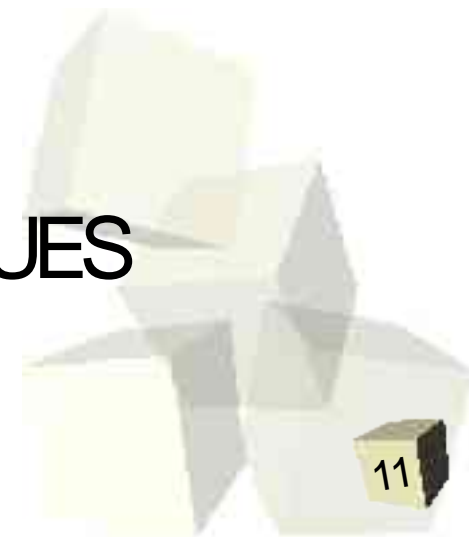
## Esempi di query SQL:

- `SELECT Nome, Cognome FROM Studente`

Nome	Cognome
Pippo	Rossi
Pluto	Verdi

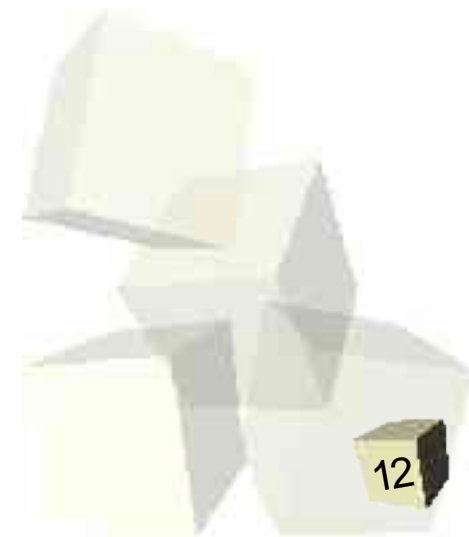
- `INSERT INTO Studente (Nome, Cognome) VALUES ('Gigi', 'Bianchi')`

Matricola	Nome	Cognome
1	Pippo	Rossi
2	Pluto	Verdi
3	Gigi	Bianchi



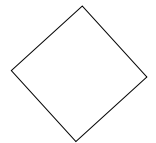
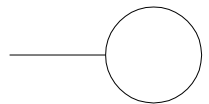
La progettazione di una base di dati deve essere **documentata e pianificata.**

- Specifiche  
*cosa deve fare*
- Diagramma Entity Relationship e sua documentazione  
*esporre l'idea*
- Diagramma logico e sua documentazione  
*come viene realizzata l'idea*



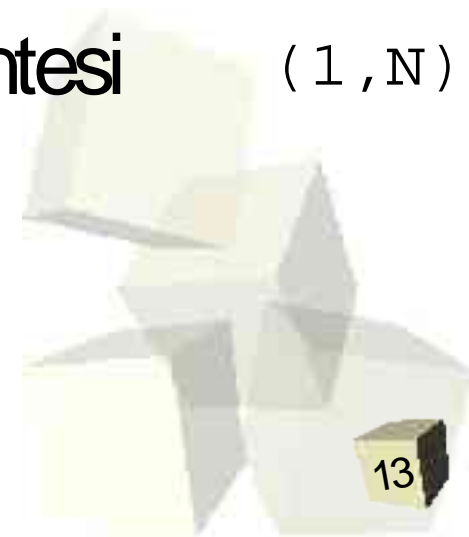
Modello concettuale, un'astrazione a oggetti della base di dati, indipendente dal sistema usato.

- Entità: oggetto, ad es. libro, studente, macchina
- Attribuiti: una proprietà dell'oggetto, ad es. titolo, nome, targa
- Relazioni: associazioni tra entità, ad es. un libro è scritto da un autore
- Cardinalità: tipo di relazione, indicata tra parentesi



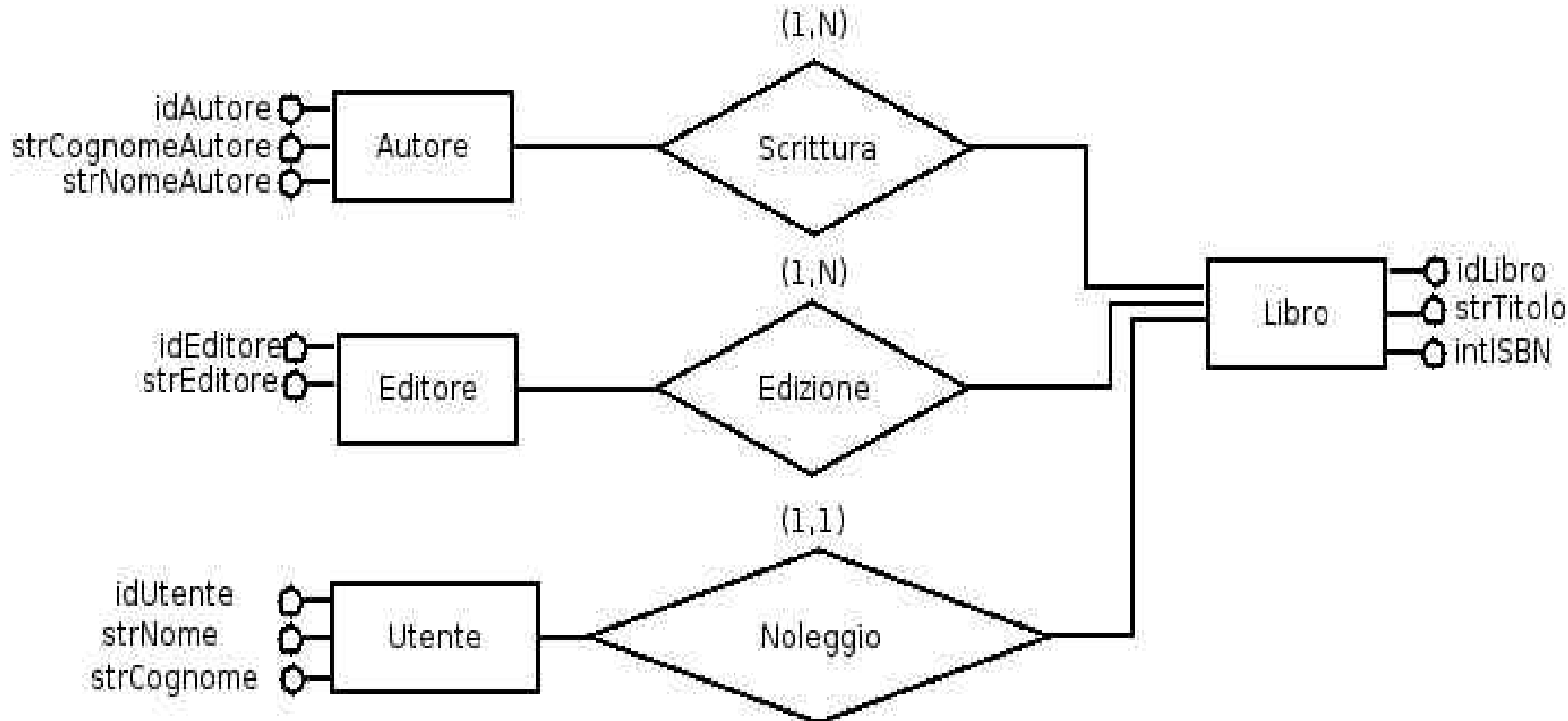
( 1 , N )

Lo scopo è ragionare sulla realtà, non sull'implementazione.



# Esempio Entity Relationship

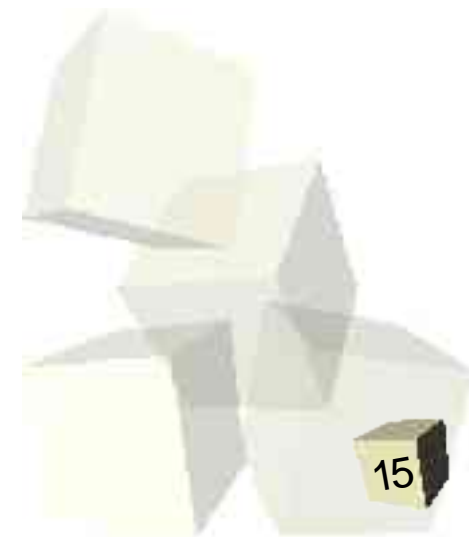
Esempio: un archivio di libri



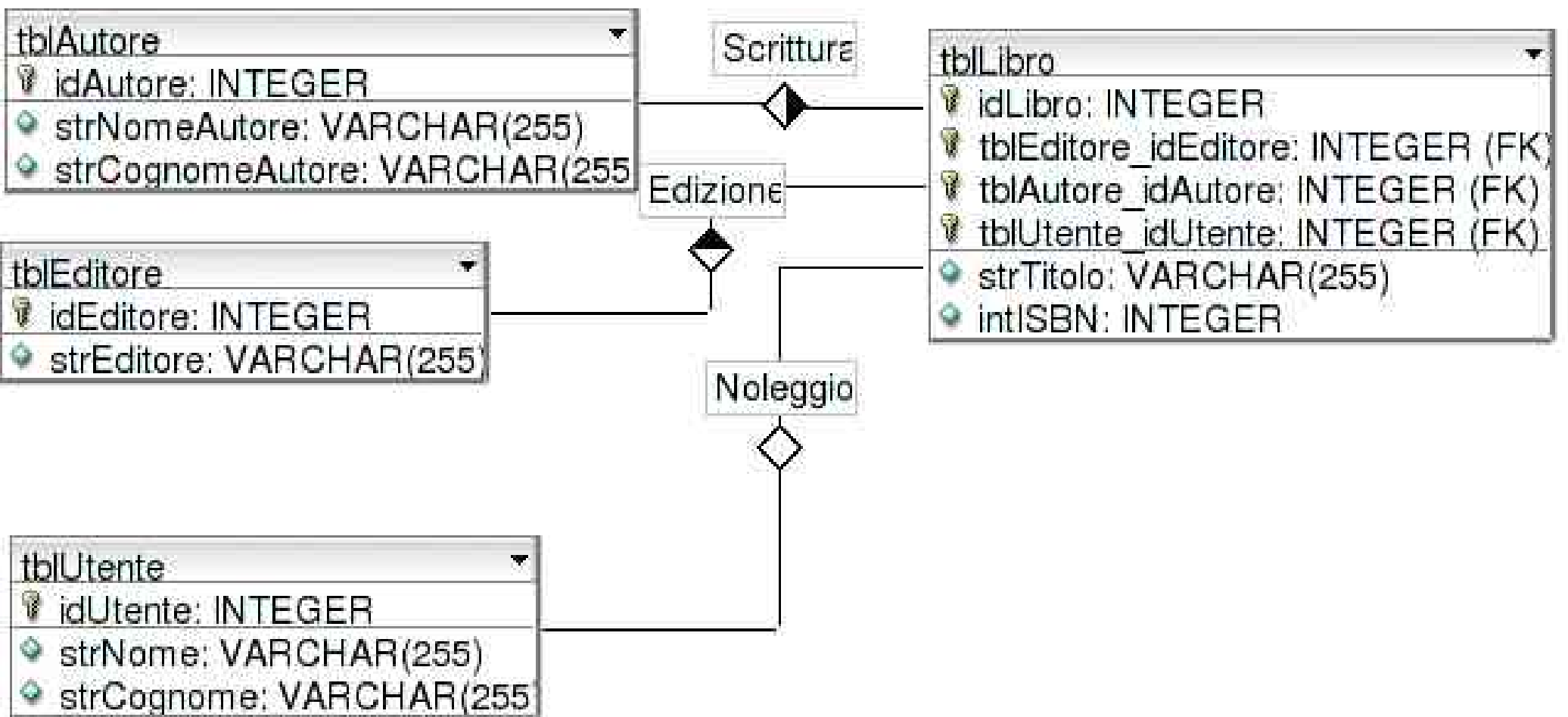
Descrizione particolareggiata della struttura delle tabelle

Specifica il **Come** viene implementato lo schema concettuale nella base di dati:

- chiavi
- vincoli d'integrità
- transazioni
- ecc.

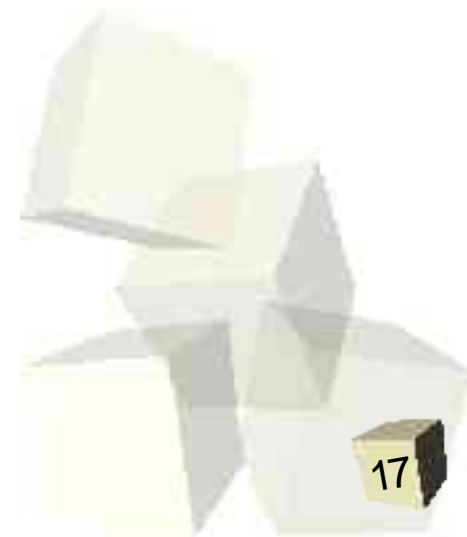


# Esempio Schema Logico



Possiamo procedere alla costruzione del database, utilizzando, ad esempio, un DBMS MySQL[1]

- di facile utilizzo
- licenza GPL
- adatto anche a piccole esigenze
- ottime prestazioni
- disponibile per Windows e per GNU/Linux



[1] <http://www.mysql.com/>

## Strumenti grafici per la progettazione e manutenzione su MySQL:

- MySQLcc [1]
- MySQLAdministrator [2]
- DBDesigner4 [3]
- MySQL Query Browser[4]

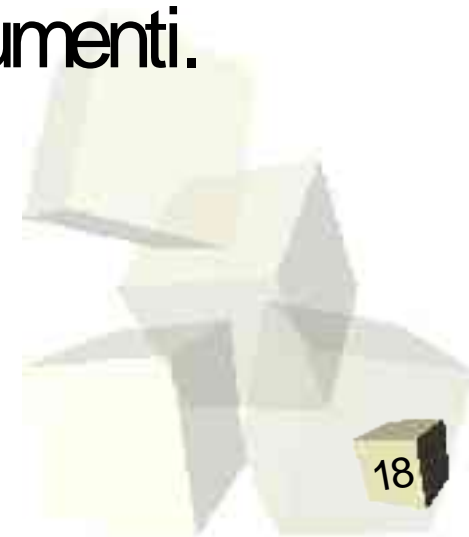
Con questi creiamo la struttura del database, ma per utilizzarlo facilmente abbiamo bisogno di altri strumenti.

[1] <http://www.mysql.com/products/mysqlcc/>

[2] <http://www.mysql.com/products/administrator/>

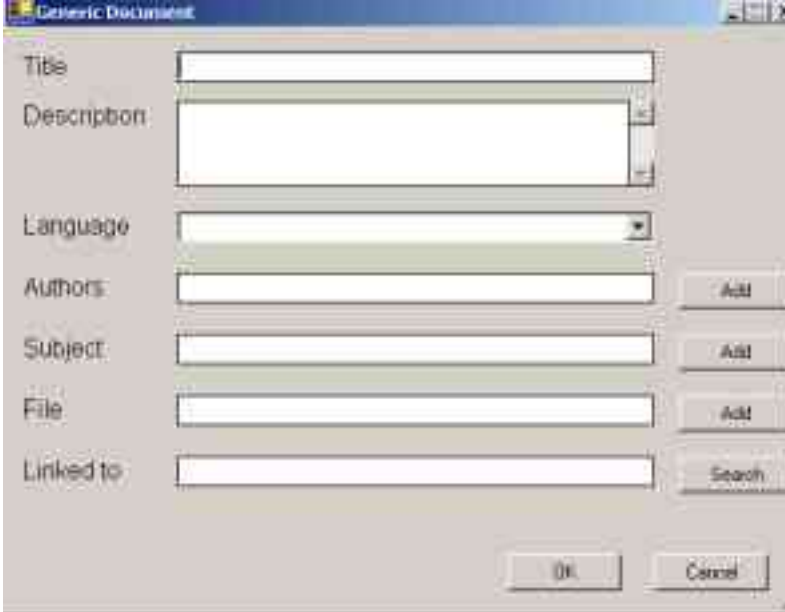
[3] <http://www.fabforce.net/dbdesigner4/>

[4] <http://dev.mysql.com/downloads/query-browser/1.1.html>

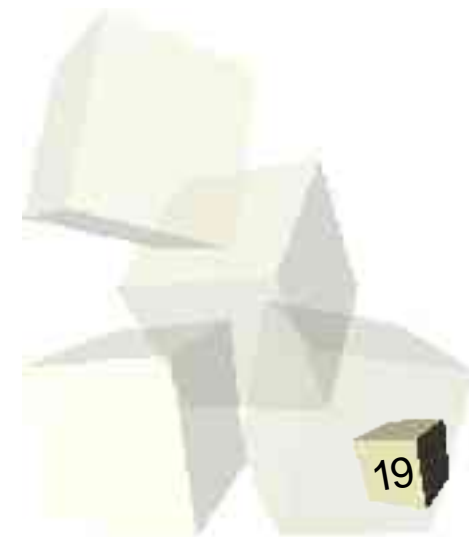


Strumento che utilizziamo per semplificare l'uso della base di dati.

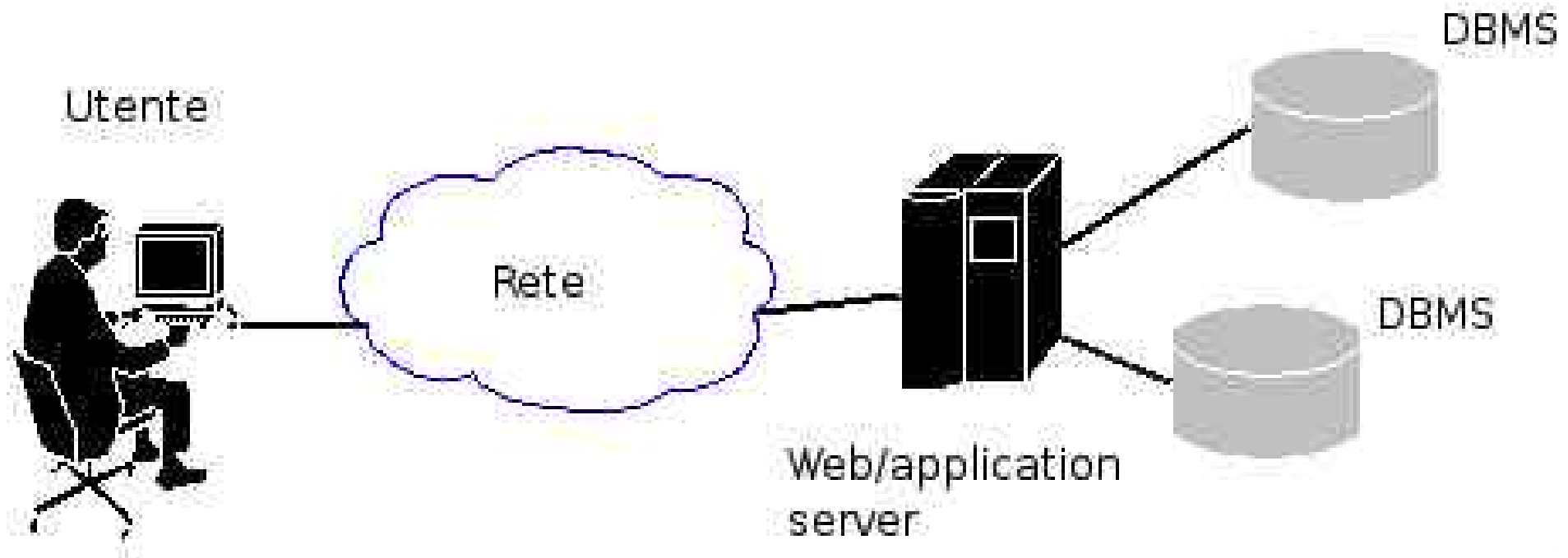
Non dobbiamo conoscere SQL o come funziona la base di dati, è tutto **trasparente** all'utente.



The image shows a screenshot of a 'Generic Document' dialog box. It contains several input fields and buttons. The fields are: 'Title' (a single-line text box), 'Description' (a multi-line text box), 'Language' (a dropdown menu), 'Authors' (a text box with an 'Add' button to its right), 'Subject' (a text box with an 'Add' button to its right), 'File' (a text box with an 'Add' button to its right), and 'Linked to' (a text box with a 'Search' button to its right). At the bottom of the dialog are 'OK' and 'Cancel' buttons.

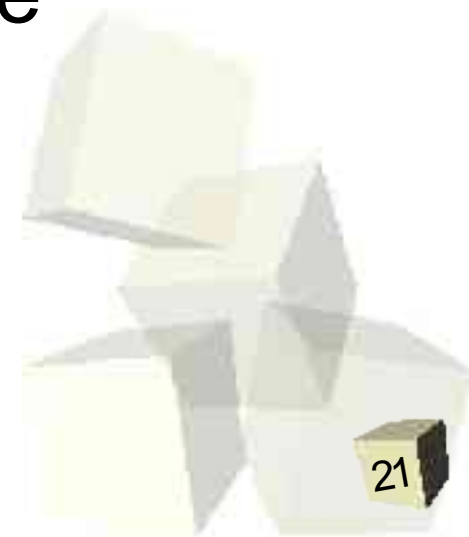


Cosa accade quando facciamo una interrogazione usando un'interfaccia:

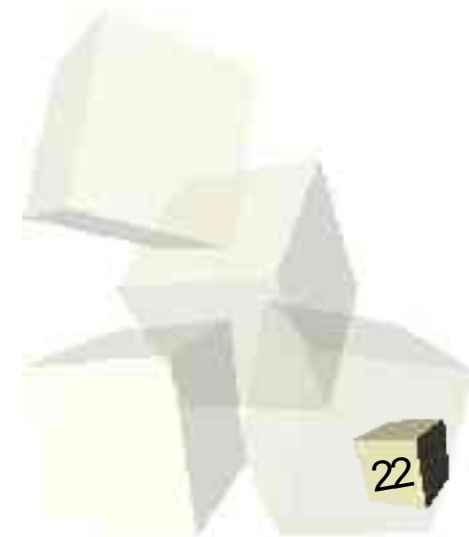


## Perché una nostra ricerca fallisce?

- x le keyword non sono corrette
  - x fare ricerche generiche, utilizzare le funzioni speciali, usare solo le parole chiavi
- x errore dell'interfaccia
  - x riempire tutti i campi, leggere il manuale
- x errore di rete
  - x riprovare più tardi, segnalarlo
- x il software ha un bug o è scritto/progettato male



- le basi di dati sono le fondamenta di un sistema informativo
- forniscono un metodo efficace per l'archiviazione di qualsiasi tipo di dato
- possiamo utilizzarle senza conoscenze tecniche con l'ausilio di strumenti grafici



- “SQL” M. J. Hernandez, Mondadori
- “Basi di dati, modelli e linguaggi di interrogazione” P. Atzeni, McGraw-Hill
- “Basi di dati, architetture e linee di evoluzione” P. Atzeni, McGraw-Hill
- “MySQL” P. DuBois, Pearson
- “SQL The Complete Reference” J. R. Groff, McGraw-Hill
- “Fundamentals of database systems” Elmasri/Navathe, Addison Wesley



Domande





Documento distribuito con licenza Creative Commons  
Attribution-NonCommercial-ShareAlike 2.0 [1]

[1] <http://creativecommons.org/licenses/by-nc-sa/2.0/legalcode>

