

Basi di dati e DBMS

Software applicativo

Basi di dati (Database, DB)

- Una delle applicazioni informatiche più utilizzate, ma meno conosciute dai non informatici
- Dal punto di vista dell'utente: avete già interagito con DB!
 - anagrafe
 - biblioteca
 - banca
 - voti degli esami
 - prenotazioni voli, treni, cinema, concerti, ospedale ...



- Cosa sta dietro?:
Sistema informatico in grado di **memorizzare grandi quantità di informazioni** in modo da facilitare le operazioni di **aggiornamento** e **il reperimento dei dati**



Sistemi informativi e basi di dati

- Dal punto di vista delle organizzazioni
- Ogni **organizzazione** ha bisogno di memorizzare e mantenere informazioni specifiche, per riuscire a raggiungere in modo efficace i propri obiettivi.
- Per esempio:
 - Compagnia telefonica -> utenze telefoniche
 - Banca -> Conti correnti bancari
 - Università -> Studenti iscritti a un corso di laurea

Sistemi informativi e basi di dati

- I **sistemi informativi** organizzano e gestiscono le informazioni necessarie alle attività di un'organizzazione:
 - Inizialmente non erano automatizzati (es. gli archivi bancari, servizi anagrafici, archivi esami universitari sostenuti, esistono da tempo)
 - Diffusione dell'informatica → **gestione automatica dei dati** → basi di dati
 - Informazione memorizzata in modo rigoroso

Informazione e dati

- Nei sistemi informatici le informazioni necessarie all'attività di un'organizzazione vengono rappresentate per mezzo di **dati** che hanno bisogno di essere **interpretati** per veicolare le informazioni
- Distinzione dato - informazione in senso tecnico
 - **Dati** = elementi di informazione costituiti da simboli che devono essere elaborate, che di per sé non hanno interpretazione
 - Mario Rossi → *nome e cognome, ma di chi?*
 - 10150 → *numero matricola? Numero di abitanti di una città? CAP? Numero di telefono?*

Informazione e dati

- Distinzione tra informazione e dati:
 - **Dati + interpretazione = informazione**
 - Domanda: “Chi è il responsabile dell’ufficio tecnico e qual è il suo numero di telefono?” →
 - Nel **contesto di questa domanda** i dati acquistano significato e accrescono la nostra conoscenza
 - Risposta: Mario Rossi, 10150 → *dati*
 - Domanda + risposta: informazione
 - Nei sistemi informatici per la gestione dei dati, al centro ho **i dati**, che vengono memorizzati nel tempo, ma l’applicazione fornisce anche un **contesto interpretativo** che rende i dati **informazioni utili per gli utenti**

Dati e applicazioni

- I dati costruiscono un patrimonio prezioso da sfruttare e proteggere
- I dati possono **variare nel tempo**
 - Es. importo conto corrente
- le **modalità** con cui i dati sono **rappresentati** in un sistema sono di solito **stabili**
 - Es. di solito una valuta è un numero con due cifre decimali
- Le **operazioni** sui dati e le procedure che operano sui dati **variano** spesso
 - Es.: ricerche
- Di conseguenza e' utile separare i dati dalle applicazioni che operano su essi

Dati strutturati e dati non strutturati

- È utile che i **dati siano strutturati** per facilitare il **reperimento** e l'**elaborazione** delle informazioni
- Per esempio, con il file di testo seguente non ho modo di ottenere esclusivamente i libri il cui autore sia Dante Alighieri: cercando “Dante” ottengo anche l’opera di Boccaccio

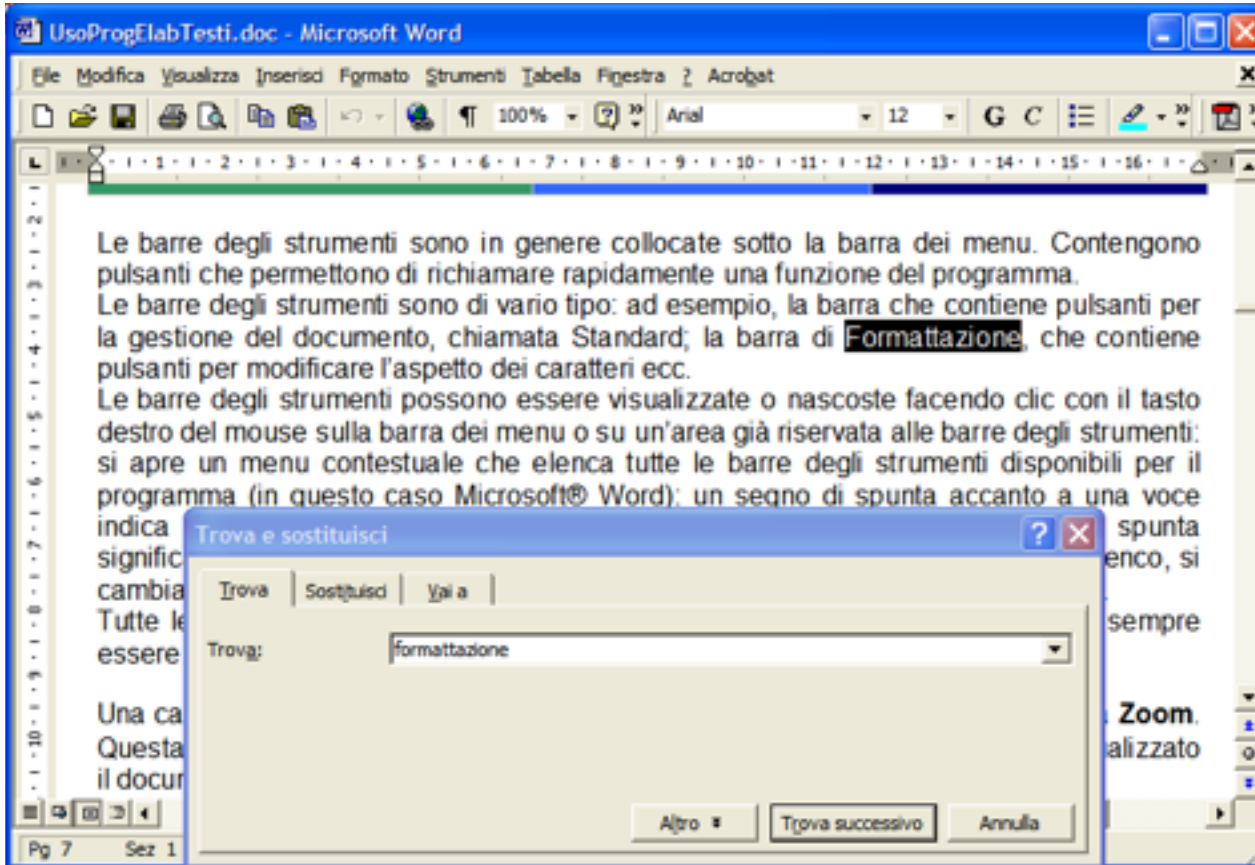
Giovanni Boccaccio, Vita di **Dante** / Giovanni Boccaccio - \S.l. : s.n., s.d.! (L'Aquila : G.T.E.) - 67 p. ; 24 cm

Dante Alighieri, La divina commedia di Dante Alighieri con tauole in rame. Tomo primo .-quarto - Firenze : nella Stamperia all'insegna dell'Ancora, 1817-1819 - 4 v ((Tutti i front. sono inc))

Dante Alighieri, La vita nuova e il canzoniere / Dante Alighieri ; a cura di Luigi Di Benedetto - XLVIII, 179 p. ; 18 cm - Collezione di classici italiani con note - 47

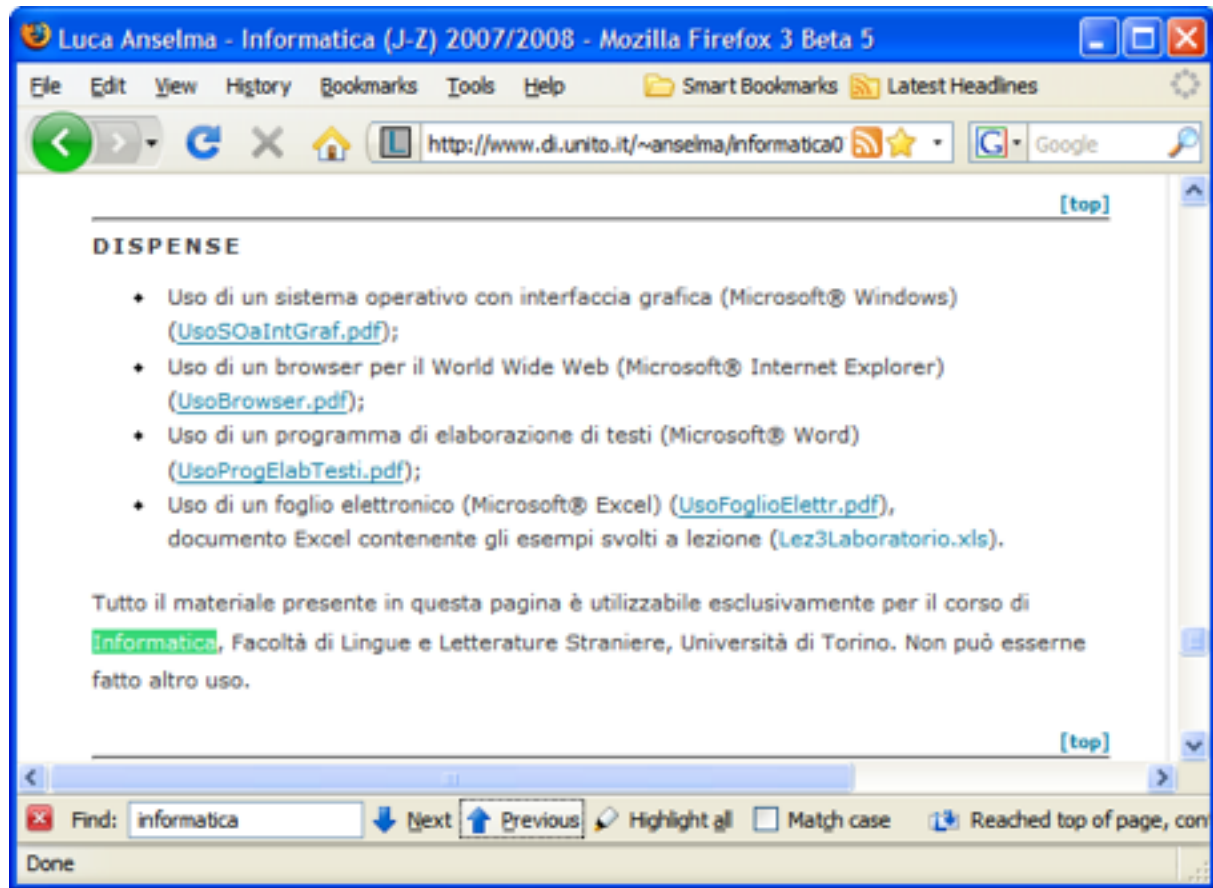
Dati non strutturati

Ricerca in un documento di testo



Dati non strutturati

Ricerca in una pagina web



Dati non strutturati

Cartella clinica

CLINICAL RECORD	NARRATIVE SUMMARY	
DATE OF ADMISSION	DATE OF DISCHARGE	NUMBER OF DAYS HOSPITALIZED
10-30-67	11-18-67	19
<p>The patient was a <u>10 year old</u> Caucasian male who was described by his parents as being very healthy other than being a nervous shy child. The only past illnesses have been measles and mumps without history of encephalitis. The <u>first abnormality</u>, now in retrospect, regarding <u>muscle control</u> was approximately 10 months ago when the patient began having difficulty with his penmanship. This was brought up by his teacher but readily apparent to the boy's parents. This was described as a tremulous abnormality with posture of the hand in extreme flexion producing indistinct letter formation. The boy is an A and B student, but not such was thought to the problem of writing. <u>On 8 August 1967</u> the patient limped down the stairs holding his right foot in inversion with plantar flexion with no history of trauma or no complaints of muscle cramps or pain. The patient was seen that day at Portsmouth Naval Station where he was told that he had muscle spasm and treated with pills, but to no avail. On 28 August the patient was admitted to</p>		

Dati strutturati

Cartella clinica

Gestione ricovero - DRG Lazio - Tariffa regionale - Tariffa fuori regione

Reparto: (Tubi)

R.O. 2006.43 **Bianchi Alfredo** nato il 23/04/2006 età 0 acc 11/12/2006

Anagrafica [F2]

Cognome CAP
Nome Città Prof.
Nato il Età M F Posiz. Prof.
Nato a Telefoni
F12] C.F.
Identificativo CRA Stato civile Medico b. Aes
Indirizzo Coniugato p. Deceduto Sì/No
Residenza Istruzione Privacy

Accettazione [F3]

1ª cartella R.O.

Data Ora Diagnosi ingr.
Causa Diagnosi istal.
Onere Ente U.I.S.
Motivo Medico acc. Alai
Tipo Medico inv. Aes
Provenienza N° P.S. Priorità
Istituto prev. Grado ass. ric. Classe ASA
note Data prenotazione Consenso anagrafica

Ingresso [F4]

Reparto Piano - Stanza - Letto

Page 20

Dati strutturati



Esempio: ricerca in PsychNET dell'American Psychological Association

The screenshot displays the APA PsycNET search interface. At the top, the logo for APA PsycNET™ and the American Psychological Association is visible, along with a 'Login' link. Below the logo is a navigation bar with tabs for 'SEARCH', 'BROWSE', 'TERM FINDER', and 'MY PsycNET'. Underneath this is another set of tabs for search methods: 'Easy Search', 'Advanced Search', 'Citation Finder', 'Cited References', 'Recent Searches', and 'My List'. The main search area is titled 'LOOK FOR' and contains two search rows. The first row has a text input field containing 'Erikson' and a dropdown menu set to 'Author'. The second row has a text input field and a dropdown menu set to 'Journal Title'. Between these rows is an 'and' connector. To the right of the search rows are links for 'Add a row' and 'Clear form', and a prominent red 'SEARCH' button. Below the search area are two sections for filtering results. The first section, 'ONLY SHOW CONTENT WHERE', has a 'Hide' button and two rows of filters, each with a dropdown menu set to 'None selected' and an 'is' connector. The second section, 'DATE', also has a 'Hide' button and two sub-sections: 'Published:' with radio buttons for 'All years to present' (selected) and 'From [] to []', and 'Added to PsycNET:' with a radio button for 'in the last [7] days'.

Basi di dati

- DB: “magazzini” di dati
- Un DB è un grande *insieme di dati* organizzati e memorizzati in forma **strutturata e omogenea**
 - **Strutturata:** es. anagrafe: nome, cognome, data di nascita, ...
 - **Omogenea:** es. anagrafe: per ogni persona mantengo le stesse informazioni

Basi di dati

- DB adatti a memorizzare dati omogenei che devono essere strutturati
 - Ad esempio:
 -  : dati anagrafe
 -  : testo di un libro
- DB gestiti da *DBMS* (DataBase Management System, *sistema software di gestione di basi di dati*)

Vantaggi dei DBMS

- Permettono di considerare i dati come risorsa di un'organizzazione
 - Una risorsa comune: a disposizione di molteplici utenti e applicazioni (anche distribuite su un territorio)
- Offrono un modello formale della realtà di interesse
 - Preciso, riutilizzabile
- Controllo centralizzato dei dati
 - Riduzione di ridondanze e inconsistenze
- Indipendenza dei dati
 - Sviluppo di applicazioni flessibili e modificabili

Svantaggi dei DBMS

- Complessi, costosi, hanno specifici requisiti in termini di software e hardware
- Difficile separare servizi utili da quelli inutili
- Alcuni sono inadatti alla gestione di poche informazioni condivise da un numero basso di utenti

Basi di dati

- Un DBMS gestisce insiemi di dati:
 - grandi
 - persistenti
 - condivisi

garantendo

- affidabilità
- privacy
- efficienza
- efficacia

Le basi di dati sono ... grandi

- **Dimensioni:**

un DB può avere dimensioni di decine di Terabyte:
migliaia di **miliardi** di byte

→ un DBMS deve prevedere una gestione dei dati
sulla memoria secondaria



Le basi di dati sono ... persistenti

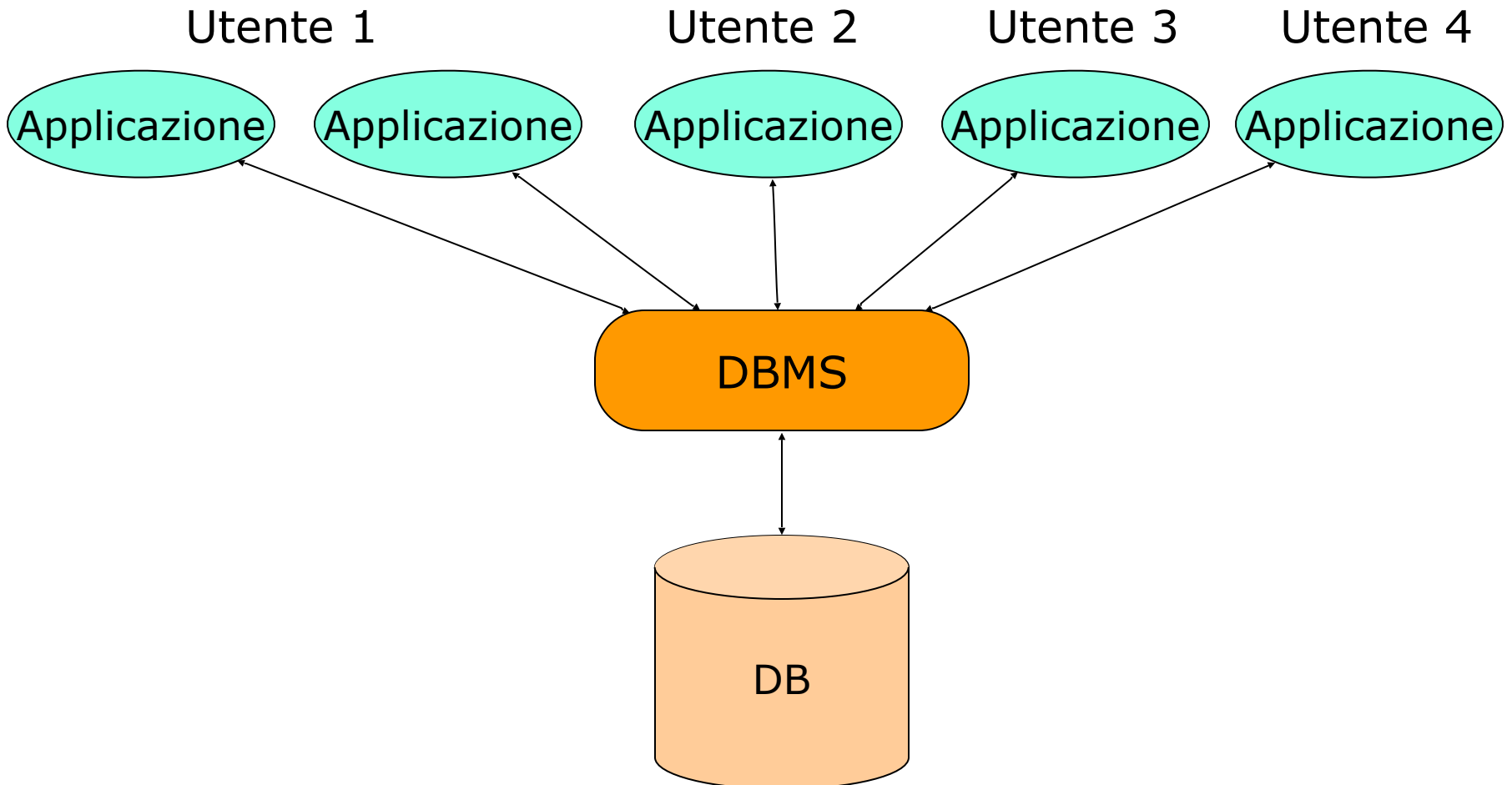
- **Persistenza:**
i dati hanno un tempo di vita che non è limitato a quello delle singole esecuzioni delle applicazioni
→ un DBMS deve gestire i dati sulla memoria secondaria



Le basi di dati sono ... condivise

- **Condivisione:** diverse applicazioni e utenti devono poter accedere a dati comuni
- L'accesso si deve svolgere secondo opportune modalità
 - Evitare le ridondanze (varie copie degli stessi dati)
 - Aggiornamenti agevoli
 - Evitare le inconsistenze dovute agli **accessi concorrenti** -> **controllo di concorrenza**

Le basi di dati sono ... condivise



Le basi di dati sono ... condivise

- Una buona gestione della condivisione permette di evitare ridondanza e incoerenza
- Ridondanza: **informazioni ripetute**
- Incoerenza: errori di “allineamento” dei dati

Le basi di dati sono ... condivise

- **Condivisione** → **concorrenza**
- **Concorrenza**: nello stesso momento più applicazioni possono accedere al medesimo dato
 - tali accessi non devono interferire tra loro per garantire l'integrità dei dati
- **Esempio**:
 - se due assegni vengono incassati contemporaneamente sullo stesso c/c, non bisogna "trascurarne" nessuno
- I DB forniscono meccanismi per gestire la concorrenza e regolamentare gli accessi

I DBMS garantiscono ... affidabilità

- **Affidabilità:**
il DBMS garantisce il contenuto del DB in caso di malfunzionamenti hardware o software
- I DBMS offrono backup e recovery;
- Capacità di ripristinare i dati (almeno parzialmente)

I DBMS garantiscono ... privatezza

- **Privatezza:**

gli utenti svolgono solo determinate azioni sui dati, con meccanismi di **autorizzazione**

Es. **biblioteca:**

- il lettore ha diritto di lettura e ricerca dei dati, ma non di modifica/ inserimento
- il bibliotecario ha diritto di modificare i dati: deve aggiungere/togliere libri e segnalare i prestiti

Es. sistema informatico per la **gestione di esami universitari**

- lo studente ha il diritto di iscriversi agli appelli, leggere il voto assegnato e accettarlo/rifiutarlo
- Il docente ha diritto di modificare i dati inserendo le date degli appelli, i voti, etc..

I DBMS garantiscono ... efficienza

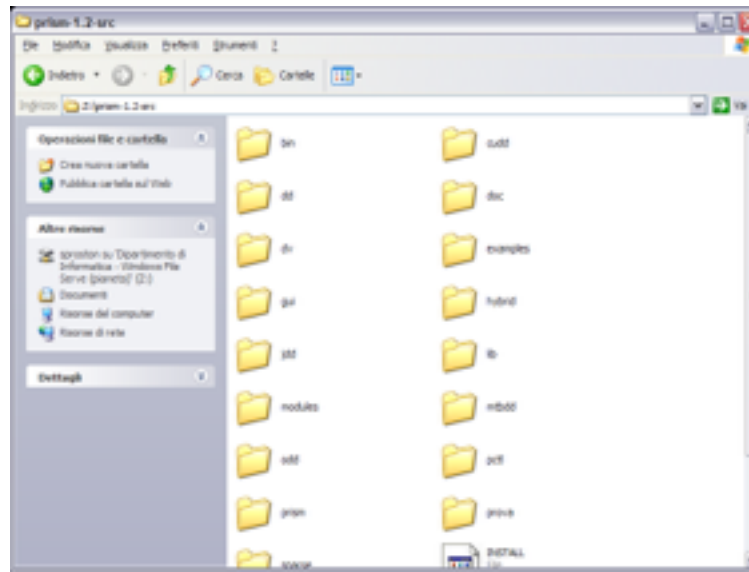
- **Efficienza:**
le operazioni devono essere svolte in tempi accettabili per l'utente (molto brevi!) nonostante la grande mole di dati
- tempi di risposta e occupazione spazio accettabili (dipende molto dalla tecnica di memorizzazione dei dati)

I DBMS garantiscono ... efficacia

- **Efficacia:**
cercano di rendere produttiva l'attività dell'**utente** e delle **organizzazioni**, offrendo funzionalità articolate, potenti e flessibili

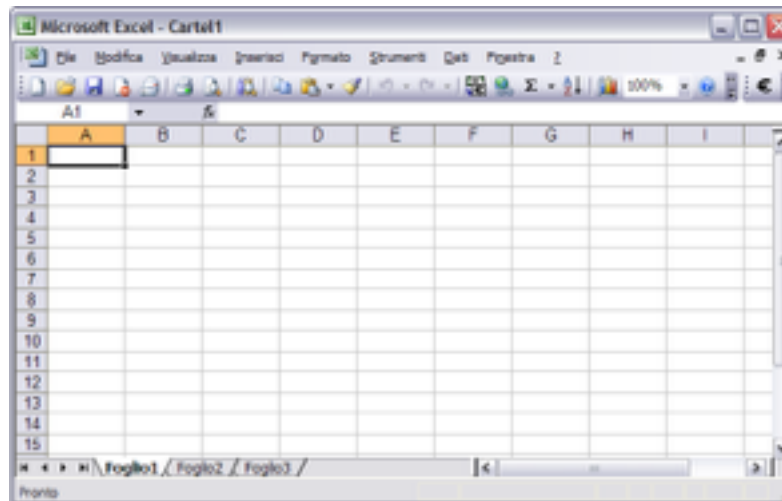
Basi di dati

- Perché non usare direttamente il file system (magari in file di testo)?
- Svantaggi:
 - Organizzazione dei file a carico dell'utente (categorizzazione dei dati per poterli reperire)
 - Dati non strutturati: come confrontarli e elaborarli?
 - Quando si hanno molti dati, i file diventano difficili da gestire



Basi di dati

- Perché non usare un foglio elettronico
 - Per esempio, i file di Excel
- Svantaggi:
 - Difficile **condivisione** tra più utenti
 - Difficile **reperire informazioni**
- Quindi: **basi di dati**



Modelli dei dati

- **Modello dei dati:** insieme di concetti utilizzati per organizzare i dati di interesse e descriverne la struttura
- Due tipi principali di modelli:
 - Logico
 - Concettuale

Modello logico dei dati

- Sottintende una specifica *rappresentazione dei dati* (tabelle, alberi, grafi, oggetti ...)
- Descrive i dati a un livello intermedio, tra ciò che vede l'utente e il livello dell'implementazione;
- La struttura dei dati è descritta in modo che risulti comprensibile da un'elaboratore
- Molte proposte in DBMS commerciali



Utente

Modello
logico

Implemen-
tazione

Ricapitolando

- Dati vs Informazioni

Modello logico dei dati

- Modello *gerarchico* (anni '60) = struttura gerarchica (albero)
- Modello *reticolare* (inizio anni '70) = struttura a grafo
- Modello *relazionale* (fine anni '70) = struttura a tabelle; permette di organizzare dati come insiemi di record a struttura fissa;
- Modello *orientato agli oggetti* (anni '80) = struttura a classi/oggetti

Modello logico dei dati

- Modello *gerarchico* (anni '60) = struttura gerarchica (albero)
- Modello *reticolare* (inizio anni '70) = struttura a grafo
- Modello *relazionale* (fine anni '70) = struttura a tabelle
- Modello *orientato agli oggetti* (anni '80) = struttura a classi/oggetti

Per esempio:
Microsoft
Access, MySQL, ...

Modello relazionale

- Introdotto nel 1970
- Affermato negli anni '80
- Attualmente è il modello di DB più diffuso: probabilmente ogni DB che incontrerete sarà relazionale
- Basato sul **concetto di relazione**
 - Relazione = rappresentazione di entità complesse tramite *attributi*
- Come nei fogli elettronici, sono organizzati in **tabelle**
 - Es.: **tabella Studenti** del database di una università

Matricola	Cognome	Nome	Sesso	DataDiNascita	LuogoDiNascita	Domicilio
300000	Rossi	Mario	M	01/01/1987	Torino	v. Fiore 2, Torino
300001	Verdi	Chiara	F	01/02/1987	Roma	v. Casa 31, Roma
300002	Rossi	Daria	F	01/01/1987	Torino	v. Fiore 2, Torino
300003	Cancelli	Chiara	F	10/10/1987	Milano	v. Tetti 1, Torino
300004	Votantonio	Raffaella	F	10/02/1985	Roma	p. Cane 1, Roma

Modello relazionale: struttura dei dati

- Le **entità** (oggetti della realtà di interesse a cui si riferisce la base di dati) **compongono le righe** (record)
- Le **caratteristiche delle entità** sono le (intestazioni delle) colonne, dette **attributi** (campi)
- Il **valore di un attributo** per una specifica entità viene inserito nella cella corrispondente
- Tutti i record di una tabella hanno lo **stesso formato**, cioè gli stessi attributi

Matricola	Cognome	Nome	Sesso	DataDiNascita	LuogoDiNascita	Domicilio
300000	Rossi	Mario	M	01/01/1987	Torino	v. Fiore 2, Torino
300001	Verdi	Chiara	F	01/02/1987	Roma	v. Casa 31, Roma
300002	Rossi	Daria	F	01/01/1987	Torino	v. Fiore 2, Torino
300003	Cancelli	Chiara	F	10/10/1987	Milano	v. Tetti 1, Torino
300004	Votantonio	Raffaella	F	10/02/1985	Roma	p. Cane 1, Roma

Modello relazionale: struttura dei dati

- Una relazione (o tabella) e' caratterizzata da
 - **Insieme di righe**: ogni riga rappresenta i dati relativi a un oggetto della realtà di interesse
 - **Insieme di colonne, attributi**
 - I dati vengono inseriti nelle celle in corrispondenza del relativo attributo
- Dati relativi agli studenti

nome relazione

Studente

colonna/attributo

record/riga/tupla

Matricola	Cognome	Nome	Sesso	DataDiNascita	LuogoDiNascita	Domicilio
300000	Rossi	Mario	M	01/01/1987	Torino	v. Fiore 2, Torino
300001	Verci	Chiara	F	01/02/1987	Roma	v. Casa 31, Roma
300002	Rossi	Daria	F	01/01/1987	Torino	v. Fiore 2, Torino
300003	Cancelli	Chiara	F	10/10/1987	Milano	v. Tetti 1, Torino
300004	Votantonio	Raffaella	F	10/02/1985	Roma	p. Cane 1, Roma

dato atomico

Es. Relazioni tra tabelle

studente

MATR	NOME	CITTA'	C-DIP
123	Carlo	Bologna	Inf
415	Paola	Torino	Inf
702	Antonio	Roma	Log

esame

MATR	COD-CORSO	DATA	VOTO
123	1	7-9-97	30
123	2	8-1-98	28
702	2	7-9-97	20

corso

COD-CORSO	TITOLO	DOCENTE
1	matematica	Barozzi
2	informatica	Meo

Query

- Per reperire le informazioni di interesse da un DB, un utente non può semplicemente leggere le tabelle:
 - Le tabelle sono molto grosse
 - Può essere necessario utilizzare più tabelle contemporaneamente
- Si usano le **query (interrogazioni)**

Query

- Una **query** permette di specificare:
 - Cosa cercare all'interno del DB (es. **criteri di selezione**, espressi attraverso operatori di confronto)
 - Quali informazioni (**campi**) visualizzare
- Una query **crea una nuova tabella** temporanea che comprende **i campi e i record di interesse**

Query: esempio

Voglio avere tutti gli esami superati dagli studenti
insieme ai dati degli studenti

Esami

CodiceCorso	Voto	Data	Matricola
L0014	24	20/02/2007	300002
L0014	28	10/02/2007	300003
L0016	27	10/09/2006	300004
L0155	25	15/01/2007	300001
L0507	30	01/06/2006	300001
L0507	21	10/09/2006	300003

Studenti

Matricola	Cognome	Nome	Sesso	DataDiNascita	LuogoDiNascita	Domicilio
300000	Rossi	Mario	M	01/01/1987	Torino	v. Fiore 2, Torino
300001	Verdi	Chiara	F	01/02/1987	Roma	v. Casa 31, Roma
300002	Rossi	Daria	F	01/01/1987	Torino	v. Fiore 2, Torino
300003	Cancelli	Chiara	F	10/10/1987	Milano	v. Tetti 1, Torino
300004	Votantonio	Raffaella	F	10/02/1985	Roma	p. Cane 1, Roma

CodiceCorso	Voto	Data	Matricola	Cognome	Nome	Sesso	DataDiNascita	LuogoDiNascita	Domicilio
L0155	25	15/01/2007	300001	Verdi	Chiara	F	01/02/1987	Roma	v. Casa 31, Roma
L0507	30	01/06/2006	300001	Verdi	Chiara	F	01/02/1987	Roma	v. Casa 31, Roma
L0014	24	20/02/2007	300002	Rossi	Daria	F	01/01/1987	Torino	v. Fiore 2, Torino
L0014	28	10/02/2007	300003	Cancelli	Chiara	F	10/10/1987	Milano	v. Tetti 1, Torino
L0507	21	10/09/2006	300003	Cancelli	Chiara	F	10/10/1987	Milano	v. Tetti 1, Torino
L0016	27	10/09/2006	300004	Votantonio	Raffaella	F	10/02/1985	Roma	p. Cane 1, Roma

La tabella risultante è generata dinamicamente

Query

- Le query possono essere create in vari modi
- Due “paradigmi”:
 - **Basato su esempi (QBE)**
 - Uso di **maschere predefinite** di interrogazione che costruiscono la query automaticamente
 - Utilizzato da utenti poco esperti
 - **Linguaggi di interrogazione (ad esempio, SQL)**
 - Vero e proprio **linguaggio (testuale)** strutturato
 - Usato dai programmi e dagli utenti più esperti

Esempi

Credits: Massimo Franceschet

Consideriamo la seguente tabella teatro:

teatro

<u>nome</u>	città	email
CSS	Udine	css@gmail.com
Litta	Milano	litta@gmail.com
Piccolo	Milano	piccolo@gmail.com
Eliseo	Roma	eliseo@gmail.com

L'interrogazione più semplice che si possa scrivere è la seguente:

```
select *  
from teatro
```

Il risultato è l'intera tabella teatro. La prima riga dell'interrogazione è detta **clausola select** e serve per selezionare le colonne della tabella che ci interessano. L'**operatore *** permette di selezionare tutte le colonne. La seconda riga dell'interrogazione è detta **clausola from** e serve per indicare quali tabelle usare. La clausola select e quella from sono obbligatorie in una interrogazione.

```
select città  
from teatro
```

Il risultato è la seguente tabella:

città
Udine
Milano
Milano
Roma

Per eliminare la duplicazione delle città (appare 2 volte Milano) si può utilizzare la parola chiave *distinct*

Clausola Where

```
select nome  
from teatro  
where città = 'Milano'
```

Il risultato è la seguente tabella:

La clausola *where* definisce un predicato sui valori degli attributi delle tabelle selezionate. Le righe che soddisfano il predicato, cioè per le quali il predicato è vero, vengono inserite nella tabella risultato.

Clausola Where

```
select nome  
from teatro  
where città = 'Milano'
```

Il risultato è la seguente tabella:

nome
Litta
Piccolo

Operatori Booleani e di Confronto

- *and or e not: Operatori Booleani*
- *=, <>, <, >, <=, >= Operatori di Confronto*

dipendente

cf	nome	cognome	dataDiNascita	stipendio
ELSDLL72	Elisa	D'Allarche	1972-04-29	2500
FRNDPP76	Fernanda	D'Ippoliti	1976-03-11	2100
MRCDLL70	Marco	Dall'Aglio	1970-01-09	2700

Il nome, il cognome e lo stipendio dei dipendenti con uno stipendio di almeno 2500 Euro.

dipendente

<u>cf</u>	nome	cognome	dataDiNascita	stipendio
ELSDLL72	Elisa	D'Allarche	1972-04-29	2500
FRNDPP76	Fernanda	D'Ippoliti	1976-03-11	2100
MRCDLL70	Marco	Dall'Aglio	1970-01-09	2700

Domanda in linguaggio naturale: Il nome, il cognome e lo stipendio dei dipendenti con uno stipendio di almeno 2500 Euro.

Query:

select nome, cognome, stipendio

dipendente

cf	nome	cognome	dataDiNascita	stipendio
ELSDLL72	Elisa	D'Allarche	1972-04-29	2500
FRNDPP76	Fernanda	D'Ippoliti	1976-03-11	2100
MRCDLL70	Marco	Dall'Aglio	1970-01-09	2700

Domanda in linguaggio naturale: Il nome, il cognome e lo stipendio dei dipendenti con uno stipendio di almeno 2500 Euro.

Query:

```
select nome, cognome, stipendio  
from dipendente
```

dipendente

<u>cf</u>	nome	cognome	dataDiNascita	stipendio
ELSDLL72	Elisa	D'Allarche	1972-04-29	2500
FRNDPP76	Fernanda	D'Ippoliti	1976-03-11	2100
MRCDLL70	Marco	Dall'Aglio	1970-01-09	2700

Domanda in linguaggio naturale: Il nome, il cognome e lo stipendio dei dipendenti con uno stipendio di almeno 2500 Euro.

Query:

```
select nome, cognome, stipendio  
from dipendente  
where stipendio >= 2500
```

dipendente

<u>cf</u>	nome	cognome	dataDiNascita	stipendio
ELSDLL72	Elisa	D'Allarche	1972-04-29	2500
FRNDPP76	Fernanda	D'Ippoliti	1976-03-11	2100
MRCDLL70	Marco	Dall'Aglio	1970-01-09	2700

Domanda in linguaggio naturale: Il nome, il cognome e lo stipendio dei dipendenti con uno stipendio di almeno 2500 Euro.

Query:

```
select nome, cognome, stipendio  
from dipendente  
where stipendio >= 2500
```

nome	cognome	stipendio
Elisa	D'Allarche	2500
Marco	Dall'Aglio	2700

dipendente

cf	nome	cognome	dataDiNascita	stipendio
ELSDLL72	Elisa	D'Allarche	1972-04-29	2500
FRNDPP76	Fernanda	D'Ippoliti	1976-03-11	2100
MRCDLL70	Marco	Dall'Aglio	1970-01-09	2700

Il nome e il cognome dei dipendenti nati dopo il 1975.

dipendente

cf	nome	cognome	dataDiNascita	stipendio
ELSDLL72	Elisa	D'Allarche	1972-04-29	2500
FRNDPP76	Fernanda	D'Ippoliti	1976-03-11	2100
MRCDLL70	Marco	Dall'Aglio	1970-01-09	2700

Il nome e il cognome dei dipendenti nati dopo il 1975.

```
select nome, cognome  
from dipendente  
where dataDiNascita > '1975-12-31'
```

Interrogazioni di congiunzione (join)

teatro

<u>nome</u>	<u>città</u>	<u>email</u>
CSS	Udine	css@gmail.com
Litta	Milano	litta@gmail.com
Eliseo	Roma	eliseo@gmail.com

dipendente

<u>cf</u>	<u>nome</u>	<u>cognome</u>	<u>dataDiNascita</u>	<u>stipendio</u>
ELSDLL72	Elisa	D'Allarche	29/04/1972	2500
FRNDPP76	Fernanda	D'Ippoliti	11/03/1976	2100
MRCDLL70	Marco	Dall'Aglio	09/01/1970	2700

lavoro

<u>teatro</u>	<u>dipendente</u>	<u>ruolo</u>
CSS	ELSDLL72	relazioni
Litta	FRNDPP76	finanza
Eliseo	FRNDPP76	controllo
Eliseo	MRCDLL70	direzione

Supponiamo di voler recuperare il nome e cognome di tutti i dipendenti del teatro CSS. Queste informazioni sono distribuite in due tabelle: dipendente e lavoro. Occorre dunque congiungere queste due tabelle mediante la chiave esterna dipendente della tabella lavoro:

Interrogazioni di congiunzione (join)

teatro

<u>nome</u>	<u>città</u>	<u>email</u>
CSS	Udine	css@gmail.com
Litta	Milano	litta@gmail.com
Eliseo	Roma	eliseo@gmail.com

dipendente

<u>cf</u>	<u>nome</u>	<u>cognome</u>	<u>dataDiNascita</u>	<u>stipendio</u>
ELSDLL72	Elisa	D'Allarche	29/04/1972	2500
FRNDPP76	Fernanda	D'Ippoliti	11/03/1976	2100
MRCDLL70	Marco	Dall'Aglio	09/01/1970	2700

lavoro

<u>teatro</u>	<u>dipendente</u>	<u>ruolo</u>
CSS	ELSDLL72	relazioni
Litta	FRNDPP76	finanza
Eliseo	FRNDPP76	controllo
Eliseo	MRCDLL70	direzione

```
select nome, cognome
from lavoro, dipendente
where (teatro = 'CSS') and (dipendente =cf)
```

Supponiamo di voler recuperare il nome e cognome di tutti i dipendenti del teatro CSS. Queste informazioni sono distribuite in due tabelle: dipendente e lavoro. Occorre dunque congiungere queste due tabelle mediante la chiave esterna dipendente della tabella lavoro:

Interrogazioni di congiunzione (join)

teatro

nome	città	email
CSS	Udine	css@gmail.com
Litta	Milano	litta@gmail.com
Eliseo	Roma	eliseo@gmail.com

dipendente

cf	nome	cognome	dataDiNascita	stipendio
ELSDLL72	Elisa	D'Allarche	29/04/1972	2500
FRNDPP76	Fernanda	D'Ippoliti	11/03/1976	2100
MRCDLL70	Marco	Dall'Aglio	09/01/1970	2700

lavoro

teatro	dipendente	ruolo
CSS	ELSDLL72	relazioni
Litta	FRNDPP76	finanza
Eliseo	FRNDPP76	controllo
Eliseo	MRCDLL70	direzione

```
select nome, cognome  
from lavoro, dipendente  
where (teatro = 'CSS') and (dipendente =cf)
```

Per aumentare la leggibilità, possiamo riscrivere la query facendo precedere ad ogni attributo il nome della corrispondente tabella:

```
select dipendente.nome, dipendente.cognome  
from lavoro, dipendente  
where (lavoro.teatro = 'CSS') and (lavoro.dipendente = dipendente.cf)
```