

Sistemi Informativi T

Home Page del corso:

<http://www-db.deis.unibo.it/courses/SI-T/>

Versione elettronica: [00.Introduzione.pdf](#)

Docente

Prof. Paolo Ciaccia

DEIS - Facoltà di Ingegneria

Viale Risorgimento, 2 - 40136, Bologna

Orario di ricevimento:

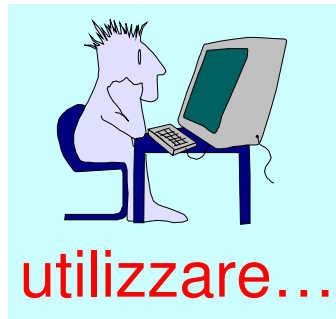
- giovedì 9:30-11
c/o palazzina IEIIT (ex-CSITE)

Contatti:

- email: **paolo.ciaccia@unibo.it**
- Web: **<http://www-db.deis.unibo.it/~pciaccia/>**

Obiettivi del corso

- Il corso intende fornire gli strumenti teorico-pratici necessari per utilizzare e progettare Basi di Dati relazionali



Orario

- Lunedì 11-13 aula 2.8
- Giovedì 11-14 aula 5.6
- Venerdì 12-14 aula 8.1

- Da circa **FINE OTTOBRE** (IV-V settimana) le ore del giovedì verranno erogate in **laboratorio**

LAB 4 (piano terra dell'edificio centrale)

L'Home Page del corso

<http://www-db.deis.unibo.it/courses/SI-T/>

Contenuti:

- Lucidi in formato PDF
- Manuali per il lab
- Testi e soluzioni delle esercitazioni in lab
- Appelli d'esame
- Testi e soluzioni di prove d'esame
- Regole d'esame
- Avvisi



Programma del corso: aula (ca. 60 ore)

- Sistemi di gestione di basi di dati (DBMS)
- Il modello relazionale dei dati
- L'algebra relazionale
- Il linguaggio SQL (Structured Query Language)
 - Definizione dei dati
 - Interrogazione e modifica dei dati
 - Trigger
 - Transazioni
- Progettazione di basi di dati (DB)
 - Il modello Entity-Relationship (E/R, o E-R)
 - Progettazione concettuale
 - Progettazione logica
 - Normalizzazione di schemi
 - Preservazione dei vincoli

Programma del corso: laboratorio (ca. 30 ore)

- Il sistema IBM DB2
 - Definizione dei dati e dei vincoli
 - Interrogazione e manipolazione dei dati
- Il sistema DB-MAIN
 - Disegno di schemi E/R
 - Progettazione concettuale
 - Progettazione logica
- Frequentare il laboratorio è importante:
 - Alcune problematiche si capiscono meglio mettendo in pratica i concetti teorici appresi a lezione
 - Acquisire manualità nell'uso di strumenti di gestione e progettazione di basi di dati è fondamentale
 - **L'esame è svolto in laboratorio!**

Testi consigliati

- Per sostenere l'esame è sufficiente il materiale reso disponibile sul sito
- Per chiunque voglia comunque avere a disposizione delle fonti in cui gli argomenti trattati a lezione vengono svolti in maniera più estesa è consigliato il testo:
 - P. Atzeni, S. Ceri, S. Paraboschi, R. Torlone. "Basi di Dati: modelli e linguaggi di interrogazione", terza edizione, McGraw-Hill Italia, 2009

Modalità d'esame

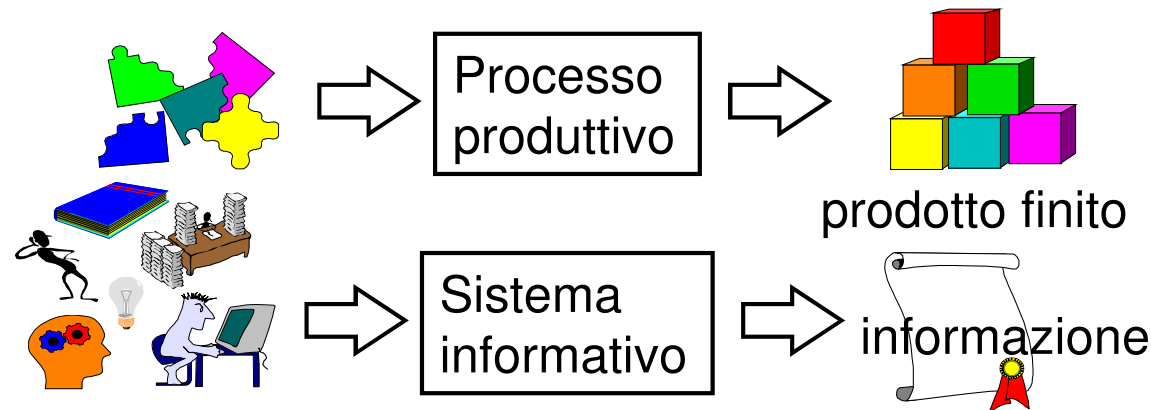
- L'esame di Sistemi Informativi T viene svolto in laboratorio
- Comprende diversi esercizi, a punteggio definito, che richiedono l'uso degli strumenti visti durante le esercitazioni (DB2 e DB-Main)
 - Informazioni più specifiche sulle modalità di calcolo del punteggio finale saranno fornite in seguito
- L'ammissione all'esame è subordinata all'iscrizione tramite [AlmaEsami](#) e all'attivazione dell'[account di facoltà](#)
- Regole generali:
 - Durante la prova non è consentito consultare testi, appunti, ecc.
 - Il **voto** acquisito in una prova ha **validità illimitata** nel tempo
 - Chi si presenta a una prova perde l'eventuale voto precedentemente acquisito
 - **Nessuna restrizione sul numero di prove** che si possono sostenere
 - La **verbalizzazione** dei voti avviene periodicamente, in date che saranno rese note sul sito del corso

Sistemi Informativi

- Un sistema informativo (SI) è un componente di una organizzazione (azienda, ente, ...) il cui scopo è gestire le informazioni utili per gli scopi dell'organizzazione stessa

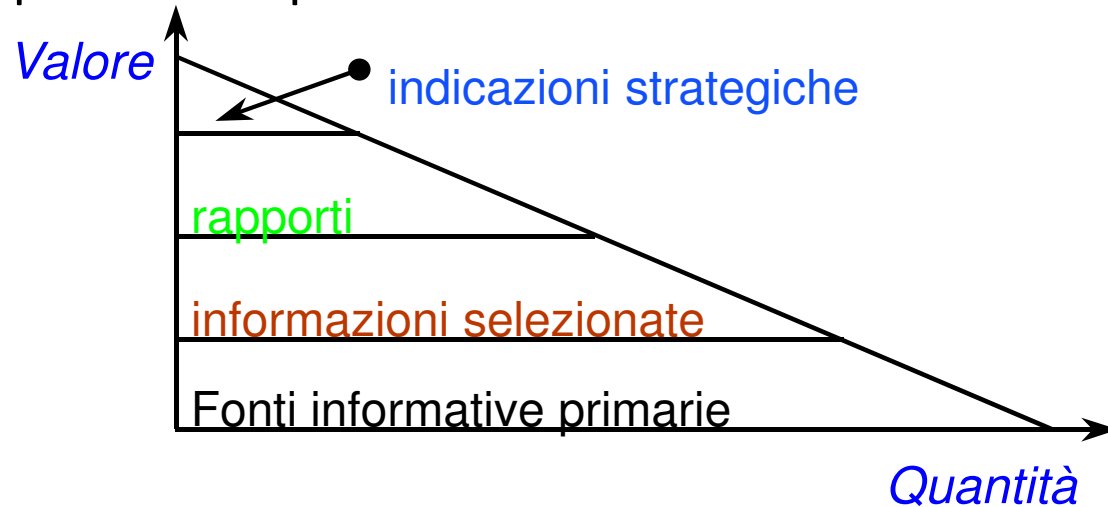
GESTIRE = acquisire, elaborare, conservare, produrre, distribuire

- L'informazione è un bene a valore crescente, necessario per pianificare e controllare con efficacia le attività dell'organizzazione, e rappresenta la materia prima che viene trasformata dai sistemi informativi, così come i semilavorati vengono trasformati dai sistemi di produzione



Il valore dell'informazione

- L'informazione è una risorsa alla stessa stregua del capitale, delle materie prime, degli impianti e delle persone, e come queste ha un costo
- È quindi importante capire il valore effettivo dell'informazione



- Il livello più basso nella scala dei valori lo occupano i “**dati grezzi**”, che possono definirsi informazione solo se interpretati nel giusto contesto

“Mario”, “Rossi” e “12345” sono **dati**

Lo **studente Mario Rossi** ha **numero di matricola 12345** è **informazione**

Dati e informazioni: un altro esempio



Lun-Ven



Sabato



Festivo

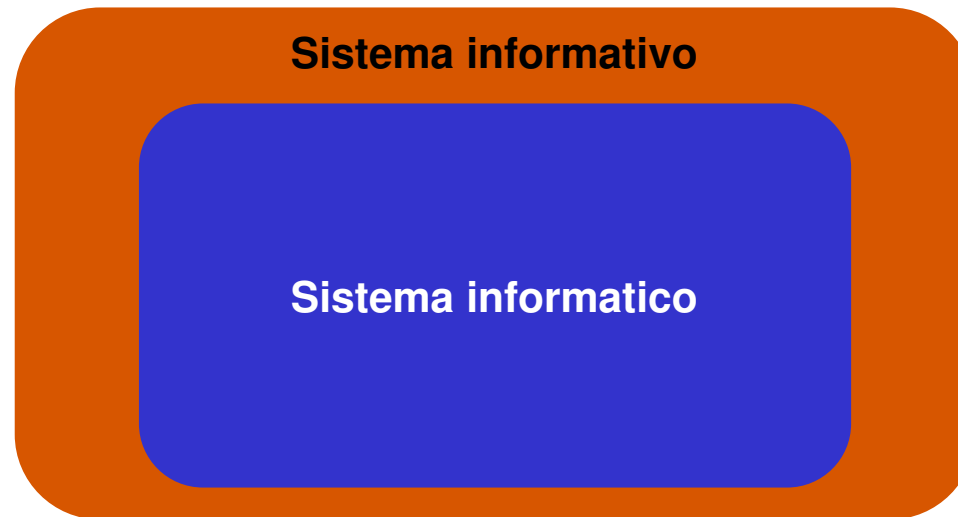
- che cosa significano questi numeri?
- cartelli stradali, in Finlandia; sono orari
- ma la differenza?
- senza "interpretazione" il dato serve a ben poco

Sistemi Informativi e Sistemi Informatici

- Un SI gestisce informazioni, ma ciò non significa necessariamente fare ricorso a strumenti automatici propri della tecnologia dell'informazione (IT)

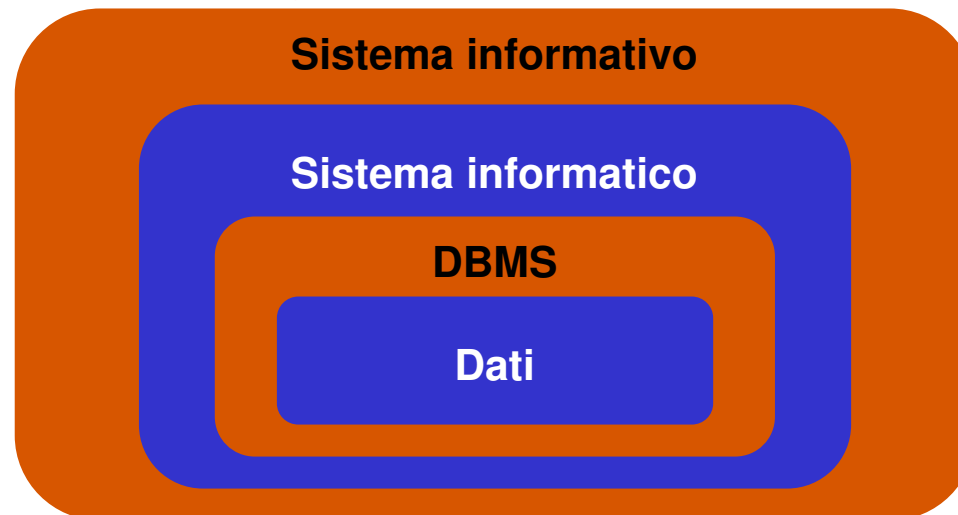
Banche e servizi anagrafici esistono da secoli!

- La parte automatizzata di un SI viene più propriamente denominata **Sistema Informatico**



Dati e Basi di Dati

- Il modo più comune con cui un sistema informatico gestisce le informazioni è attraverso la **rappresentazione codificata dei dati di interesse**
- Intuitivamente, una **Base di Dati (DB - Data Base o Database)** può pensarsi come una collezione di dati che rappresentano le informazioni di interesse per un'organizzazione
- In termini più precisi, un DB è una **collezione di dati gestita da un DBMS = Data Base Management System**



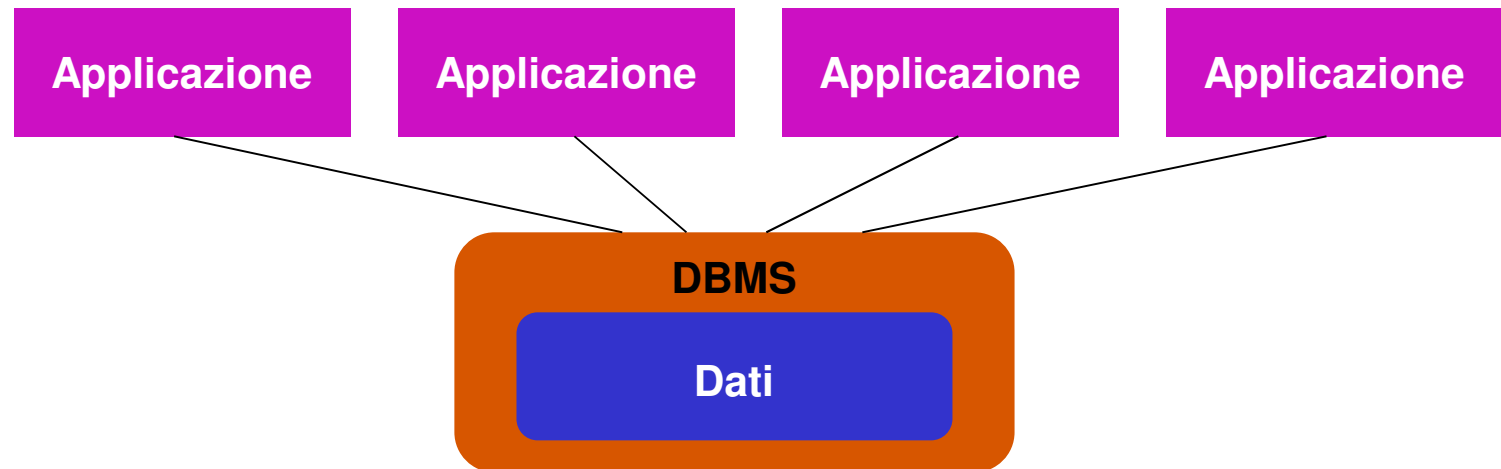
Dati e applicazioni: il ruolo dei DBMS

- In qualsiasi organizzazione sono molteplici le applicazioni del sistema informatico che utilizzano gli stessi dati (o quasi)

**Immatricolazione, RegistrazioneEsami,
DomandaLaurea, PianidiStudio, ...**

sono tutte procedure applicative che **condividono** i dati di uno studente, dei corsi di una facoltà, ecc.

- Un DBMS è un sistema software in grado di gestire collezioni di dati che sono **condivise da più applicazioni e utenti** (e molto altro ancora...)



DBMS: caratteristiche di base

- Le caratteristiche di un DBMS non si limitano ovviamente alla sola condivisione dei dati e verranno trattate nel seguito con maggior dettaglio
- ... ma è importante avere chiaro da subito che un DBMS:
 - è in grado di gestire **grandi quantità di dati** (Giga-Tera byte e oltre)
 - è in grado di garantirne la **persistenza** (anche a fronte di guasti)
 - è in grado di garantire **elevate prestazioni** (utilizzo efficiente delle risorse e ottimizzazione dei tempi di esecuzione delle operazioni)
 - offre una “visione strutturata” dei dati che dipende dal **modello (logico) dei dati** supportato

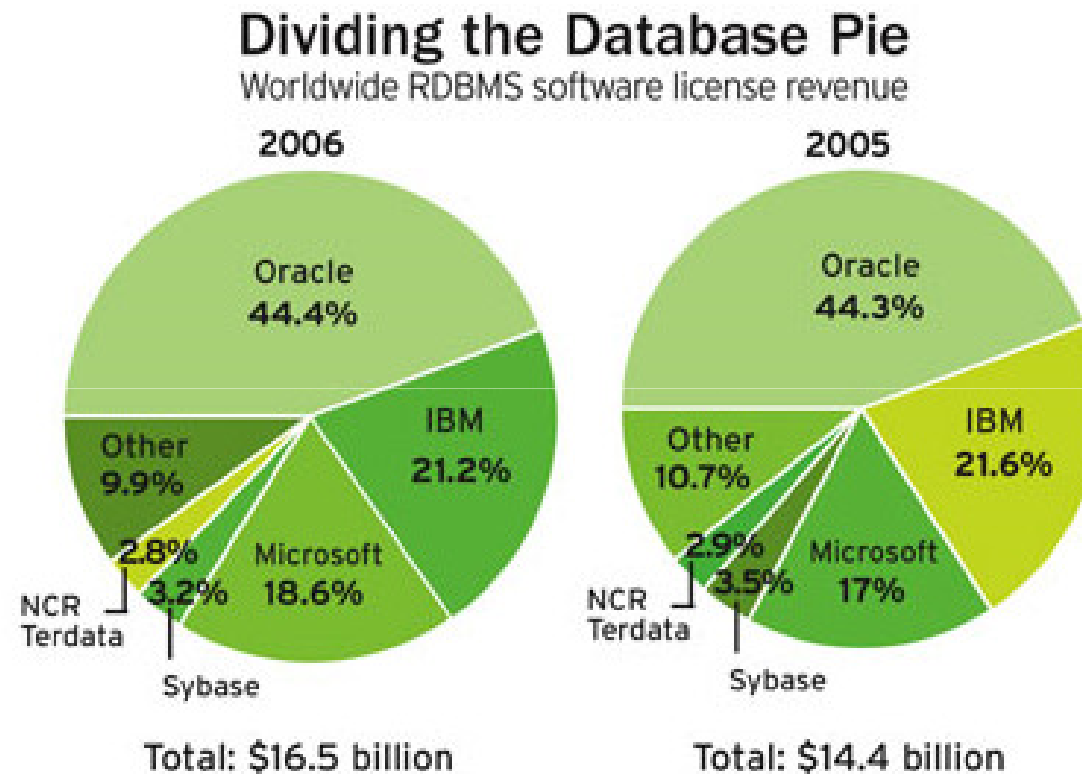
RDBMS = DBMS che supporta il **modello relazionale** dei dati

Modello relazionale \approx i dati sono rappresentati in forma tabellare

Uno sguardo al mercato dei DBMS

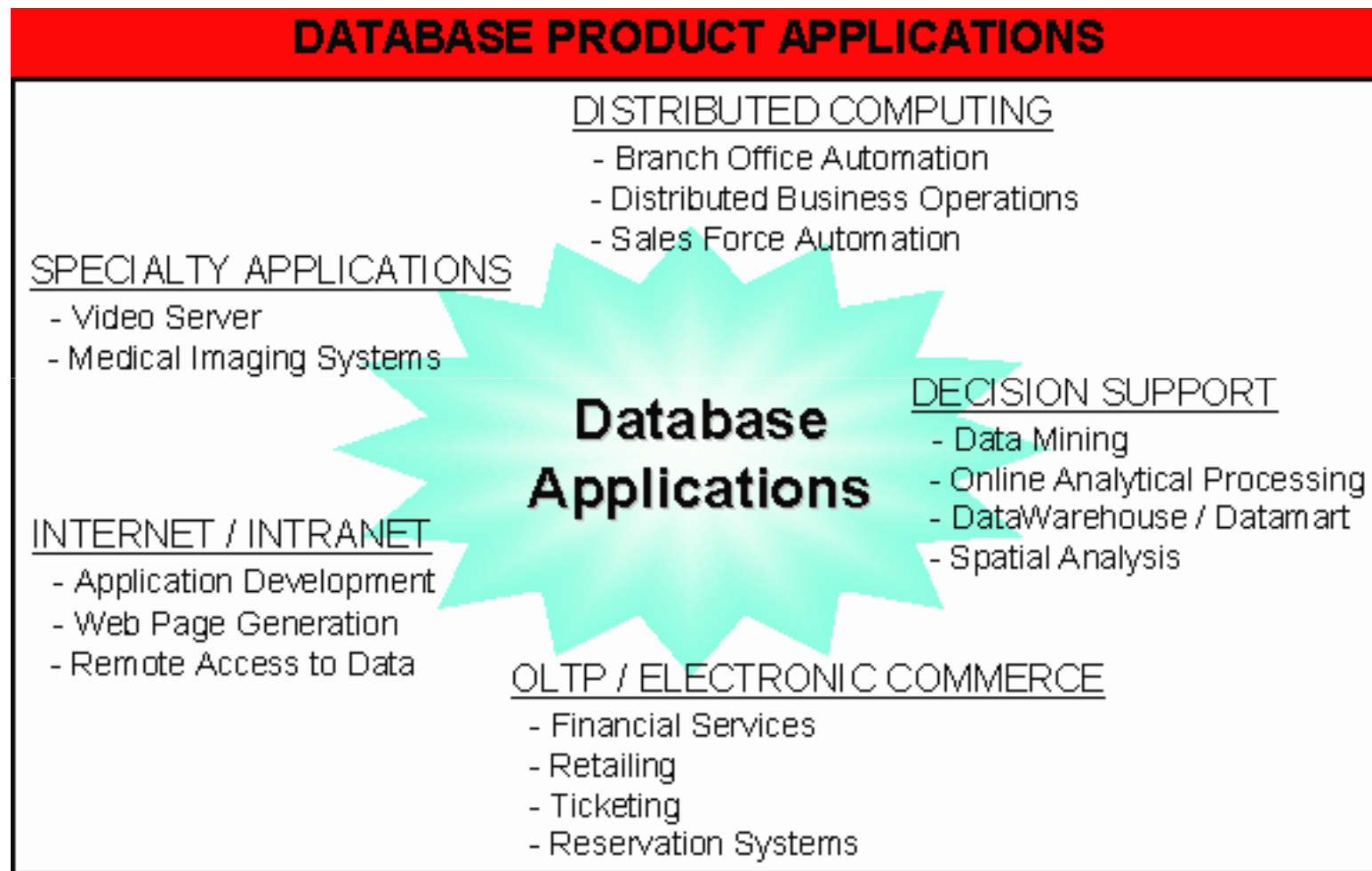
- Oggi il mercato dei DBMS (il cui fatturato si aggira oggi intorno a **18x10⁹ US\$**) è dominato da un numero ristretto di produttori:
 - Oracle
 - IBM (DB2 e altri)
 - Microsoft (SQL Server)
 - Sybase
- A questi se ne aggiungono altri del mercato “open-source”:
 - mySQL
 - PostgreSQL
 - InterBase
- Il **fatturato globale** del mercato dei DBMS (di cui **l'80% è imputabile ai RDBMS**) cresce di un fattore **> 10%** ogni anno

DBMS Market Share



Source: IDC

Applicazioni dei Data Base



Quanto sono grandi i Data Base?

Norm. Data Volume, All Environments, OLTP

Company/Organization	Norm. Data Volume (GB)	DBMS	Platform	Architecture	DBMS Vendor	System Vendor	Storage Vendor
UPS	29,329	DB2	z/OS	Federated/SMP	IBM	IBM	IBM
US Bureau of Customs & Border Protection	10,642	CA-Datcom	z/OS	Distributed/SMP	CA	IBM	Hitachi
Elsevier	7,873	Oracle RAC	UNIX	Centralized/Cluster	Oracle	Sun	IBM
Anonymous	5,761	Sybase ASE	UNIX	Centralized/SMP	Sybase	IBM	Hitachi
Turkcell	4,654	Oracle	UNIX	Centralized/SMP	Oracle	HP	EMC
AIM Healthcare Services	4,080	SQL Server	Windows	Centralized/SMP	Microsoft	IBM	EMC
Caixa Econômica Federal	3,733	CA-IDMS	z/OS	Centralized/SMP	CA	IBM	EMC
Amazon.com	2,176	Oracle RAC	Linux	Centralized/Cluster	Oracle	HP	HP
Stadtwerke Munich	1,300	Oracle RAC	Windows	Centralized/Cluster	Oracle	Fujitsu Siemens	EMC
Cellcom	1,279	Oracle	UNIX	Centralized/SMP	Oracle	HP	EMC
Coop	1,231	Oracle	UNIX	Centralized/SMP	Oracle	Sun	Hitachi

Fonte: Winter Corporation, 2005

Quante “righe” contengono i Data Base?

Number of Rows, All Environments, OLTP

Company/Organization	Number of Rows (Millions)	DBMS	Platform	Architecture	DBMS Vendor	System Vendor	Storage Vendor
UPS	89,621	DB2	z/OS	Federated/SMP	IBM	IBM	IBM
Anonymous	82,639	Sybase ASE	UNIX	Centralized/SMP	Sybase	IBM	Hitachi
Verizon Communications	50,747	SQL Server	Windows	Centralized/SMP	Microsoft	HP	EMC
Anonymous	43,549	DB2	z/OS	Centralized/SMP	IBM	Amdahl	IBM
Experian	35,398	DB2	z/OS	Centralized/SMP	IBM	IBM	EMC
Commander Communications Ltd	31,849	SQL Server	Windows	Centralized/SMP	Microsoft	Dell	EMC
LG Credit Card	30,546	DB2	z/OS	Centralized/Cluster	IBM	IBM	EMC
US Bureau of Customs & Border Protection	24,364	CA-Datcom	z/OS	Distributed/SMP	CA	IBM	Hitachi
Caixa Econômica Federal	20,582	CA-IDMS	z/OS	Centralized/SMP	CA	IBM	EMC
Turkcell	16,855	Oracle	UNIX	Centralized/SMP	Oracle	HP	EMC
AIM Healthcare Services	14,286	SQL Server	Windows	Centralized/SMP	Microsoft	IBM	EMC

Fonte: Winter Corporation, 2005

Quanto vengono usati i Data Base?

Circa 300000/sec!

Peak Workload, Mainframe and other, OLTP

Company/Organization	Peak Workload	DBMS	Platform	Architecture	DBMS Vendor	System Vendor	Storage Vendor
UPS	1,134,034,718	DB2	z/OS	Federated/SMP	IBM	IBM	IBM
US Bureau of Customs & Border Protection	340,838,403	CA-Datcom	z/OS	Distributed/SMP	CA	IBM	Hitachi
Experian	202,214,000	DB2	z/OS	Centralized/SMP	IBM	IBM	EMC
State Street Corporation	195,430,140	CA-Datcom	z/OS	Federated/SMP	CA	IBM	EMC
Caixa Econômica Federal	131,847,300	CA-IDMS	z/OS	Centralized/SMP	CA	IBM	EMC
CheckFree Corporation	66,046,711	DB2	z/OS	Centralized/Cluster	IBM	IBM	EMC
LG Credit Card	36,639,038	DB2	z/OS	Centralized/Cluster	IBM	IBM	EMC
Land Registry	6,464,623	DB2	z/OS	Centralized/Cluster	IBM	IBM	IBM
US Department of Treasury/FMS	331,509	CA-Datcom	z/OS	Centralized/SMP	CA	IBM	IBM

Peak Workload measures the peak SQL statements or database operations per hour

Fonte: Winter Corporation, 2005

Uno sguardo al corso

- Esistono (almeno) 3 “punti di vista” (ruoli) possibili:
 - **Utente**, ovvero come usare un DB
 - Richiede la conoscenza del modello dei dati, dei linguaggi supportati dal DBMS e delle modalità con cui un'applicazione può collegarsi a un DB
 - La parte di sviluppo applicazioni non è trattata in questo corso
 - **Progettista**, ovvero come progettare un DB
 - È necessario capire come i requisiti informativi di un'organizzazione complessa possono tradursi in strutture concrete
 - **Amministratore**, ovvero come amministrare un DB
 - Richiede anche conoscenze su come è fatto un DBMS (in particolare per motivi di efficienza)
 - Non trattato in questo corso
- Esistono poi anche altri ruoli, tra cui: chi progetta e sviluppa applicazioni basate su DB, chi progetta DBMS, ecc.

Il Data Base della JLUNGA

Incassi

Data	Ora	Cassa	Prodotto	Qtà	Importo
20-12-1997	17:53	21	Panettone GnamGnam 1Kg	2	26000
20-12-1997	17:53	21	Spumante Bollicin 1 lt.	5	60000
20-12-1997	18:01	21	Dentifricio WhiteTeeth	1	3400
20-12-1997	18:02	15	Spumante Bollicin 1 lt.	2	24000
21-12-1997	9:06	3	Caffè BlackMoka 250 gr.	1	4100

- Molte operazioni “semplici” di registrazione delle vendite
- Complesse operazioni di analisi delle vendite:
 - Importo complessivo delle vendite
 - Importo vendite per giorno, cassa, fascia oraria, ...
 - Correlazioni di vendita (chi compra il panettone, spesso compra anche lo spumante)



L'obiettivo è permettere alle applicazioni di specificare queste operazioni nel modo più semplice e “uniforme” possibile, e senza conoscere i dettagli dell'organizzazione **fisica dei dati**

Un esempio: le casse il 20/12/1997

■ Usando i file

```
openfile(f,`mypath/myfile`);  
for i=1 to num_casse  
    do somma[i] = 0;  
while not eof(f) do  
    read(f,rec);  
    if rec.Data = `20/12/1997`  
    then somma[rec.Cassa] +=  
        rec.Importo  
enddo;  
closefile(f);  
for i=1 to num_casse  
    do print(i,somma[i]);
```

■ Usando DBMS e SQL

```
Select Cassa,sum(Importo)  
        As Totale  
From    Incassi  
Where   Data = `20/12/1997`  
Group by Cassa;
```

Cassa	Totale
21	350265
3	234150
15	415320

Legato a:

- path e file
- formato dei dati
- elaborazione sequenziale

Un altro esempio: la cassa 21 nel 1997

```
openfile(f, `mypath/myfile`);  
for i=1 to 365*num_prodotti  
    do somma[i] = 0;  
while not eof(f) do  
    read(f, rec);  
    if rec.Cassa = 21 and  
        year(rec.Data) = 1997  
    then  
        i = converti(rec.Data,  
            rec.Prodotto);  
        somma[i] += rec.Importo  
    enddo;  
closefile(f);  
for i=1 to 365*num_prodotti  
    do print(i, somma[i]);
```

```
Select Data, Prodotto, sum(Importo)  
From Incassi  
Where Cassa = 21  
And year(Data) = 1997  
Group by Data, Prodotto  
Order by Data, Prodotto;
```

- sort implicito in "converti"
- memoria male utilizzata
- poco chiaro

... ma anche SQL può essere complesso!

- Ad esempio, se si vuole trovare

per ogni cassa, il prodotto più venduto il 27/09/2001

```
With CP (Cassa, Prodotto, SumQta)
As (Select Cassa, Prodotto, sum(Importo)
    From Incassi
    Where Data = `27/09/2001'
    Group by Cassa, Prodotto)
Select Cassa, Prodotto
From CP
Where SumQta = (Select max(CP1.SumQta)
                From CP As CP1
                Where CP1.Cassa = CP.Cassa);
```

Come usare SQL (1)

- Le istruzioni SQL possono essere eseguite interattivamente...

The screenshot shows a window titled "Editor comandi 1" with a menu bar (Editor comandi, Selezionato, Modifica, Vista, Strumenti, ?) and a toolbar. Below the toolbar are tabs for "Comandi", "Risultati dell'interrogazione", and "Plan di accesso". The "Comandi" tab is active, displaying the following SQL query:

```
SELECT LastName, Salary, Job, Birthdate
FROM Employee
WHERE Salary > 30000
ORDER BY Salary DESC
```

Below the query, the "Risultati dell'interrogazione" tab is active, showing a table of results. The table has four columns: LASTNAME, SALARY, JOB, and BIRTHDATE. The results are sorted by salary in descending order.

LASTNAME	SALARY	JOB	BIRTHDATE
HAAS	52.750,00	PRES	24-ago-1933
LUCCHESI	46.500,00	SALESREP	5-nov-1929
THOMPSON	41.250,00	MANAGER	2-feb-1948
GEYER	40.175,00	MANAGER	15-set-1925
KWAN	38.250,00	MANAGER	11-mag-1941
PULASKI	36.170,00	MANAGER	26-mag-1953
STERN	32.250,00	MANAGER	7-lug-1945

Come usare SQL (2)

- ... o inserendole nel codice di un'applicazione scritta in un linguaggio di programmazione "ospite" (ad es. Java)

```
System.out.println("Retrieve some data from the database");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("SELECT * FROM employee");
// display the result set
while (rs.next()) {
    String a = rs.getString(1);
    String str = rs.getString(2);
    System.out.print(" empno= " + a);
    System.out.print(" firstname= " + str);
    System.out.print("\n");
}
rs.close();
stmt.close();
```

Progettazione di basi di dati

- Seconda parte del corso
- Aspetti metodologici (quindi: non solo nozioni ma “**forma mentis**”)
- Ognuna delle molteplici applicazioni che usa i dati di un DBMS ha i suoi specifici **requisiti**, quindi:

Per progettare una “buona” base di dati bisogna innanzitutto capire bene che dati deve contenere e che relazioni esistono tra tali dati

- Sembra semplice...(sembra)...

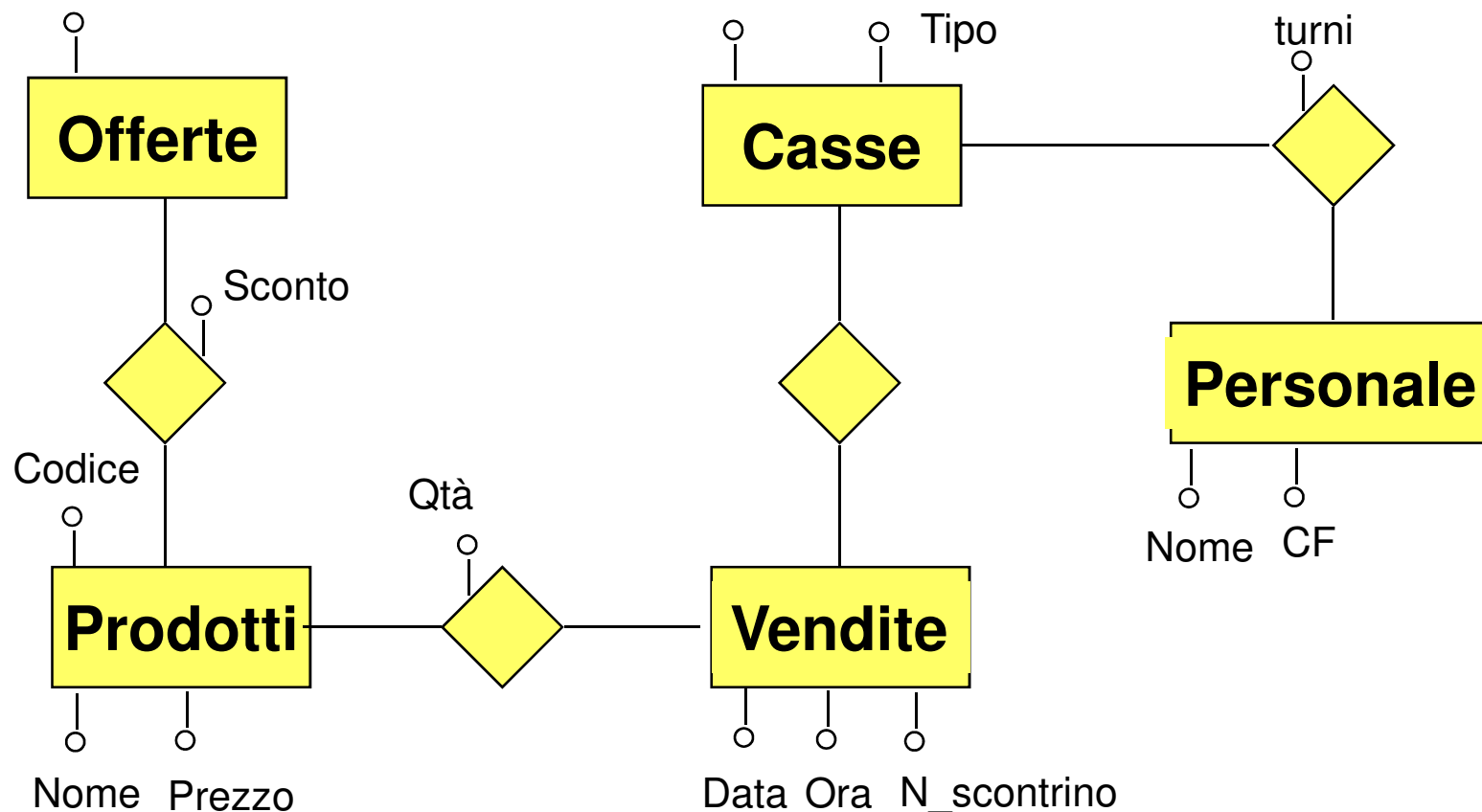
Il DB della JLUNGA si complica

- ... e deve anche gestire:
 - il personale (turni, stipendi, ecc.)
 - le offerte promozionali
 - gli ordini ai fornitori
 - ...
- Per ognuno di questi aspetti vanno analizzati i relativi requisiti



C'è comunque bisogno di strumenti efficaci, chiari e sintetici per rappresentare i dati di interesse e le loro relazioni (ovvero: il risultato dell'analisi dei requisiti)

(parte del) nuovo DB della JLUNGA



- Schema “**concettuale**” che va “**tradotto**” in tabelle (relazioni)
- In un DB reale le tabelle possono essere **centinaia-migliaia**!!

Un altro esempio di requisiti... 😊

La POOC (Prodotti Originali Oltremodo Cari) è una società che gestisce una catena di supermercati per i coloni dei satelliti di Giove (Io, Europa, Ganimede e Callisto). I prodotti della POOC sono o locali (ogni satellite ha i suoi prodotti tipici, ad esempio Ganimede è famoso per le gelatine di plancton) o periodicamente importati dalla Terra.

Ogni satellite vende agli altri i suoi prodotti secondo un meccanismo che prevede il trasferimento di grosse partite al satellite che ha fatto l'offerta migliore, il quale può poi in parte rivenderle a prezzo maggiorato (recentemente Callisto ha comprato da Io circa 1000 tonnellate di *vino gioviale* al prezzo di 5000 Crediti a tonnellata, e ne ha rivendute 300 a Ganimede ad un prezzo di 6000 Crediti).

I prodotti provenienti dalla terra, viceversa, non possono essere rivenduti, ma vengono distribuiti (allo stesso prezzo per tutti, e invariabile nel tempo per statuto della POOC) presso i supermercati POOC in modo proporzionale al numero di coloni che vi si riforniscono (al POOC "Ganimede III" fanno riferimento circa 12000 coloni, e quindi l'ultimo cargo dalla Terra vi ha depositato circa 10 milioni di uova). Un supermercato può ordinare prodotti in quantità superiore alla quota assegnata, ma per questo deve fornire in cambio alla Terra dei prodotti locali, in rapporto di volta in volta contrattato (l'ultima volta che "Callisto IV" ha voluto più prosciutto crudo, ha dovuto dare alla Terra 20 tubetti di dentifricio al plasma di fluoro per ogni etto di prosciutto).

Dati e vincoli

- In ogni situazione reale i dati che si vogliono gestire devono rispettare certi **vincoli**
- Nel DB della JLUNGA:
 - Ogni cassa ha un numero univoco
 - Ogni persona sta al più a una cassa alla volta
 - Il prezzo di un prodotto in un giorno non varia
 - ... e molti altri ancora (quali?)
- La conoscenza dei vincoli permette di:
 - Progettare un "buon" DB
 - Analizzare la "bontà" di un DB

Vincoli: un semplice esempio

Incassi

Data	Ora	Cassa	Prodotto	Qt	Importo
20-12-1997	17:53	21	Panettone GnamGnam 1Kg	2	26000
20-12-1997	17:53	21	Spumante Bollicin 1 lt.	5	60000
20-12-1997	18:01	21	Dentifricio WhiteTeeth	1	3400
20-12-1997	18:02	15	Spumante Bollicin 1 lt.	2	???
21-12-1997	9:06	3	Caff. BlackMoka 250 gr.	1	4100

- Vincolo: il prezzo di un prodotto in un dato giorno non varia
- Pertanto: il valore nella 4ª riga non è "nuova" informazione (vale 24000!)



La tabella Incassi contiene molti dati ridondanti

- La specifica dei vincoli è anche importante per evitare inconsistenze!

Ad es. se la 4ª riga contenesse il valore 30000, il vincolo sarebbe violato e quindi saremmo in presenza di un errore

Riassumiamo:

- Un **DB** è una collezione di dati (ad es. rappresentati in forma di tabelle) che rappresentano le informazioni di interesse per un'organizzazione
- Progettare un DB significa trovare un “buon modo” per strutturare i dati, che rispetti sia i requisiti propri del **Sistema Informativo** dell'organizzazione sia i **vincoli** che sui dati vengono imposti
- Un **DBMS** è un sistema software che gestisce DB
- Per eseguire operazioni su un DB è necessario scrivere istruzioni in un linguaggio (tipicamente dichiarativo: **SQL**) supportato dal DBMS