




# Basi Di Dati e di conoscenza

Modelli dei dati



# Contenuti della Lezione

- DBMS vs file system 
- Modello di rappresentazione dei dati
- Schemi ed istanze
- Architettura di un DBMS
- Indipendenza logica e fisica
- Linguaggi per basi di dati
- Attori del sistema

# DBMS vs file system

- L'efficienza di un sistema si misura (come in tutti i sistemi informatici) in termini
  - di **tempo** di esecuzione (tempo di risposta) e spazio
  - di **memoria** (principale e secondaria).
- I DBMS, a causa della varietà di funzioni, non sono necessariamente più efficienti dei file system.
- L'efficienza è il risultato della qualità del DBMS e delle applicazioni che lo utilizzano.

# DBMS vs file system

- La gestione di insiemi di dati grandi e persistenti è possibile anche attraverso sistemi più semplici — gli ordinari **file system** dei sistemi operativi
- I file system prevedono forme rudimentali di condivisione: "tutto o niente". Nei DBMS, è consentita una maggiore flessibilità
- I DBMS estendono le funzionalità dei file system, fornendo più servizi ed in maniera integrata (efficacia)

# DBMS vs file system

- Nei programmi tradizionali che accedono a file, ogni programma contiene una descrizione della struttura del file stesso, con i conseguenti rischi di incoerenza fra le descrizioni (ripetute in ciascun programma) e i file stessi.
- Nei DBMS, esiste una porzione della base di dati (**il catalogo o dizionario**) che contiene una descrizione centralizzata dei dati, che può essere utilizzata dai vari programmi.

# Contenuti della Lezione

- DBMS vs file system
- Modello di rappresentazione dei dati
- Schemi ed istanze
- Architettura di un DBMS
- Indipendenza logica e fisica
- Linguaggi per basi di dati
- Attori del sistema



# Modello di rappresentazione dei dati

- I DBMS non sono progettati per gestire un unico caso d'uso, al contrario sono software in grado di gestire dati eterogenei.
- Al fine di creare e gestire la corrispondente base di dati uno schema dei dati deve essere fornito al DBMS
- Lo schema viene costruito secondo un modello di dati ben definito. Un Modello di Dati è una collezione di costrutti usati per descrivere lo schema dei dati, le loro relazioni e i vincoli di consistenza che devono essere applicati sugli stessi.
- Tramite questo schema dei dati si fornisce al DBMS una rappresentazione dei dati, in modo tale da permettere l'organizzazione della gestione

# Modello di rappresentazione dei dati

- Le descrizioni e rappresentazioni dei dati a livelli diversi
- permettono l'indipendenza dei dati dalla rappresentazione fisica:
- i programmi fanno riferimento alla struttura a livello più alto, e le rappresentazioni sottostanti possono essere modificate senza necessità di modifica dei programmi
- Questo concetto viene realizzato tramite il **modello dei dati**

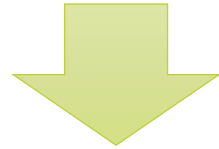


# Modello di rappresentazione dei dati

- **Modello dei dati:** Un Modello dei dati è un insieme di concetti utilizzati per organizzare i dati di interesse e descriverne la struttura in modo che essa risulti comprensibile ad un elaboratore (e non solo...).
- E anche (**Definizione di Ullman**)...
  - Un formalismo matematico composto da :
    - una notazione per descrivere i dati
    - un insieme di operazioni per manipolare tali dati

# Modello di rappresentazione dei dati

Più precisamente



**Modello di Dati:** insieme di costrutti utilizzati per organizzare i dati di interesse e descriverne la **struttura** e la **dinamica**.

- Un **modello** di dati è costituito da:
  - **Costrutti sintattici** per definire i dati
  - **Regole semantiche** per interpretare i dati
  - **Linguaggi** per manipolare i dati

# Organizzazione dei dati in una base di dati relazionale

## Orario

Nome	Cognome	Matricola	Voto medio
Mario	Rossi	1	24
Luigi	Bianchi	2	28
Rosa	Rossa	3	26

# Modelli per la rappresentazione dei dati nei DBMS

- Insieme di costrutti utilizzati per organizzare i dati di interesse e descrivere le operazioni su di essi
- Due tipi di modelli
  - **Modello logico**
  - **Modello concettuale**

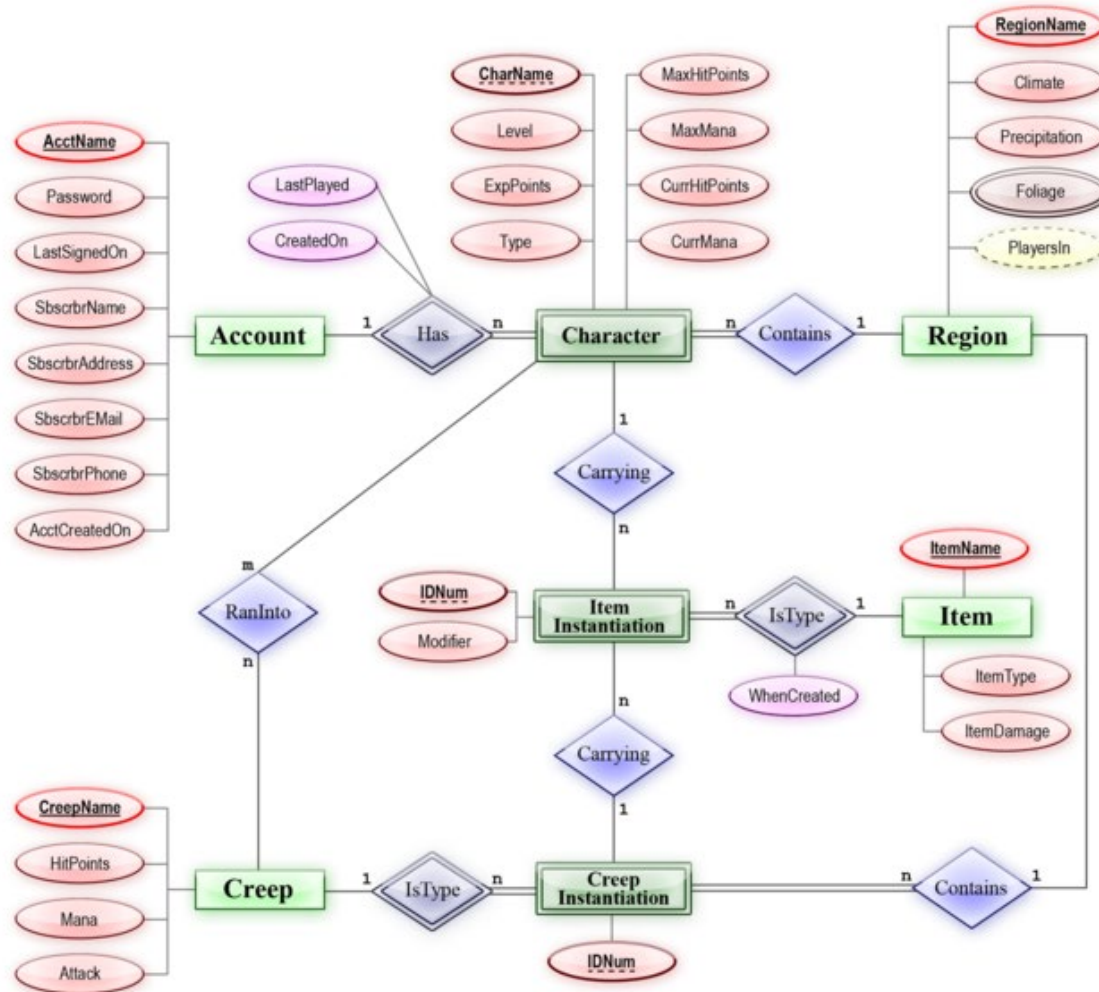
# Modello logico

- Esistono diverse tipologie di modelli logici definiti nel tempo
- Descrivono l'organizzazione dei dati nei DBMS "**visibile**" all'utente
- Sono indipendenti dalle strutture fisiche
  - **Gerarchico** e **reticolare**
    - utilizzano riferimenti espliciti (puntatori) fra record
  - **Modello ad oggetti:** (ODBMS, Object Database Management System)
    - L'informazione è rappresentata in forma di **oggetti**
    - Utilizzate in un mercato di nicchia rispetto al modello relazionale (RDBMS). (applicazioni real time)
  - **Relazionale** "è basato su valori (RDBMS).
    - anche i riferimenti fra dati in strutture (relazioni) diverse sono rappresentati per mezzo dei valori stessi

# Modello concettuale

- hanno l'obiettivo di descrivere i **concetti** del mondo reale
- utilizzati nelle fasi iniziali della progettazione
- **Entity-Relationship** e **Modello Classi Associazioni** (UML)

# Modello concettuale



# Contenuti della Lezione

- DBMS vs file system
- Modello di rappresentazione dei dati
- Schemi ed istanze
- Architettura di un DBMS
- Indipendenza logica e fisica
- Linguaggi per basi di dati
- Attori del sistema





# Schemi e istanze

- In ogni base di dati esistono:
  - lo **schema**, sostanzialmente invariante nel tempo, che ne descrive la struttura (aspetto intensionale)
    - nell'esempio, le intestazioni delle tabelle
  - l'**istanza**, i valori attuali, che possono cambiare anche molto rapidamente (aspetto estensionale)
    - nell'esempio, il "corpo" di ciascuna tabella

# Schema

## Struttura della base dati

### Orario

Nome	Cognome	Matricola	Voto medio
------	---------	-----------	------------

# Istanza

## Contenuto della base dati

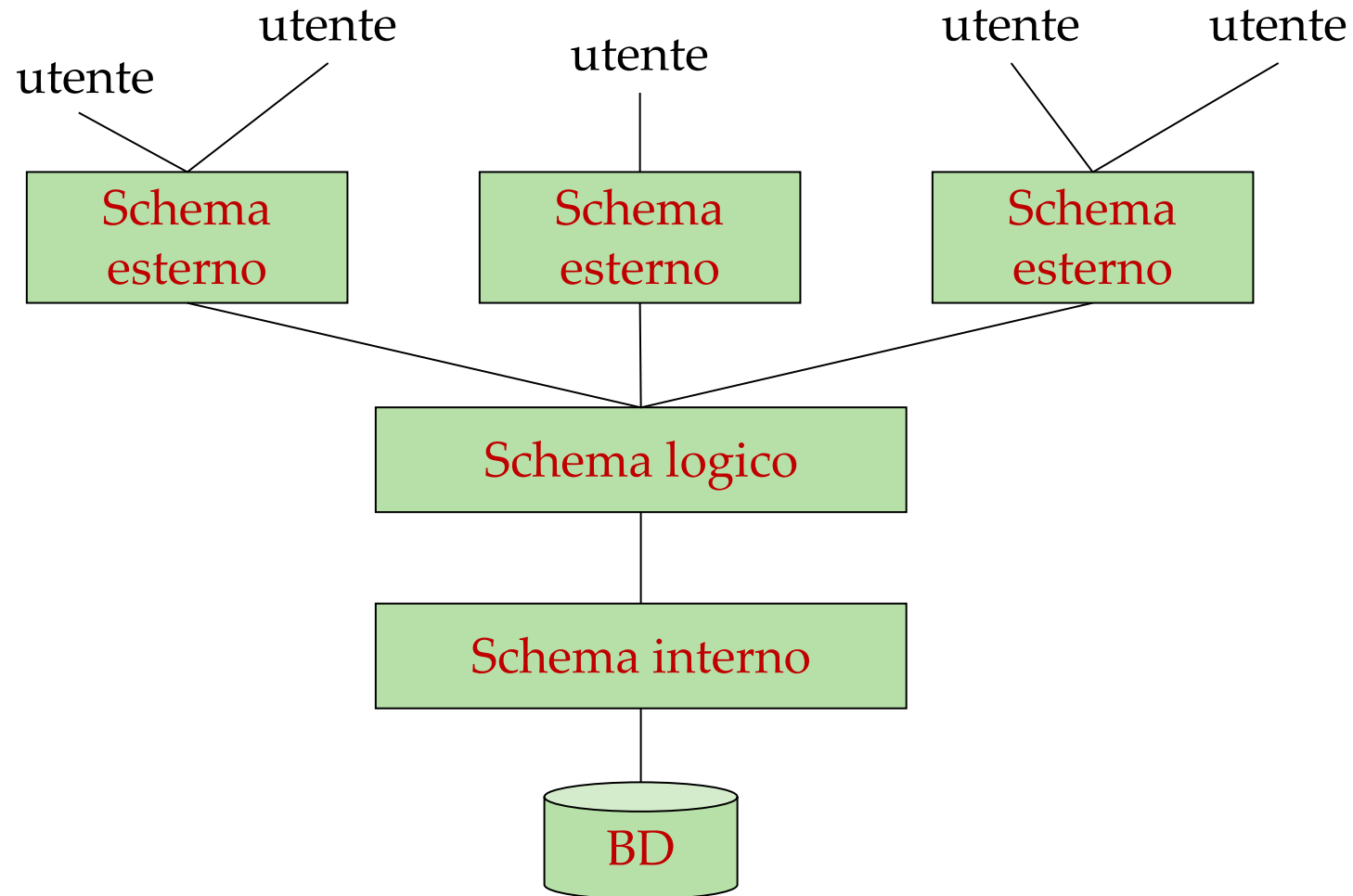
Nome	Cognome	Matricola	Voto medio
Mario	Rossi	1	24
Luigi	Bianchi	2	28
Rosa	Rossa	3	26

# Contenuti della Lezione

- DBMS vs file system
- Modello di rappresentazione dei dati
- Schemi ed istanze
- Architettura di un DBMS
- Indipendenza logica e fisica
- Linguaggi per basi di dati
- Attori del sistema



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



# Architettura di un DBMS: schemi

- **schema logico**: descrizione della base di dati nel modello logico del DBMS
- **schema fisico**: rappresentazione dello schema logico per mezzo di strutture memorizzazione (file)
- **schema esterno**: descrizione di parte della base di dati in un modello logico (“viste” parziali, derivate, anche in modelli diversi)

# Viste

## Corsi

Corso	Docente	Aula
Basi di dati	Rossi	DS3
Analisi I	Neri	N3
Fisica 2	Bruni	N3
Fisica 1	Bruni	G

## Aule

Nome	Edificio	Piano
DS1	Sogene	Terra
N3	PP2	Terra
G	Ingegneria	Primo

## CorsiSedi

Corso	Aula	Edificio	Piano
Analisi1	N3	PP2	Terra
Fisica 2	N3	PP2	Terra
Fisica1	G	Ingegneria I	Primo

# Schemi dei dati: Esempio

- Un'analogia con il mondo della programmazione facendo riferimento alle matrici.
- **Livello concettuale:**

```
int a[n][m];
```

- **Livello fisico:** (con  $a_0$ =locazione iniziale)

$A[i][j]$  è nella locazione  $a_0 + 4(m(i-1) + j - 1)$



# Schemi dei dati: Esempio

- **schema fisico**: il vettore è memorizzato in  $4(n*m)$  celle contigue a partire dalla locazione  $a_0$  (un intero è rappresentato con 4 Byte)
- **schema concettuale**: è costituito dalla dichiarazione

**`int a[n][m];`**

dove, A è una matrice di interi con n righe e m colonne.

- **Schema esterno (vista)**: un possibile schema è costituito dalla seguente funzione.

$$f(i) = \sum_{j=1}^m a[i][j]$$

# Contenuti della Lezione

- DBMS vs file system
- Modello di rappresentazione dei dati
- Schemi ed istanze
- Architettura di un DBMS
- Indipendenza logica e fisica
- Linguaggi per basi di dati
- Attori del sistema



# Indipendenza dei dati

- L'indipendenza dei dati è una conseguenza della articolazione in livelli
- l'accesso avviene solo tramite il livello esterno (che può coincidere con il livello logico)
- due forme:
  - **indipendenza fisica**
  - **indipendenza logica**

# Indipendenza fisica

- Si parla di **indipendenza fisica** quando il livello logico e quello esterno sono indipendenti da quello fisico
  - una relazione è utilizzata nello stesso modo qualunque sia la sua realizzazione fisica
  - la **realizzazione fisica** può cambiare senza che debbano essere modificati i programmi

# Indipendenza logica

- Si parla di **indipendenza logica** quando il livello esterno è indipendente da quello logico
  - aggiunte o modifiche alle viste non richiedono modifiche al livello logico
  - modifiche allo schema logico che lasciano inalterato lo schema esterno sono trasparenti

# Contenuti della Lezione

- DBMS vs file system
- Modello di rappresentazione dei dati
- Schemi ed istanze
- Architettura di un DBMS
- Indipendenza logica e fisica
- Linguaggi per basi di dati
- Attori del sistema



# Linguaggi per basi di dati

- operazioni sullo schema
  - **DDL: data definition language**
- operazioni sui dati
  - **DML: data manipulation language**

# Un'operazione DDL

(sullo schema)

```
create table orario (  
    insegnamento char(20) ,  
    docente        char(20) ,  
    aula           char(4)  ,  
    ora            char(5)  )
```



# Un'operazione DML

(sull'istanza)

```
select docente  
from orario  
where aula = 'N1'
```

# Linguaggi per basi di dati

- I DBMS dispongono di vari linguaggi e interfacce diverse
  - **linguaggi testuali interattivi (SQL)**
  - **comandi (come quelli del linguaggio interattivo)** immersi in un linguaggio **ospite** (Java, C, Spark, etc.)
  - **comandi (come quelli del linguaggio interattivo)** immersi in un linguaggio ad hoc, con anche altre funzionalità (p.es. per grafici o stampe strutturate), anche con l'ausilio di strumenti di sviluppo (p. es. per la gestione di maschere)
  - **con interfacce amichevoli** (senza linguaggio testuale)

# SQL, linguaggio interattivo

```
SELECT Corso, Aula, Piano  
FROM Aule, Corsi  
WHERE Aula = 'N3'  
AND Piano='Terra' ;
```

Corso	Aula	Piano
Analisi1	N3	Terra
Fisica 2	N3	Terra

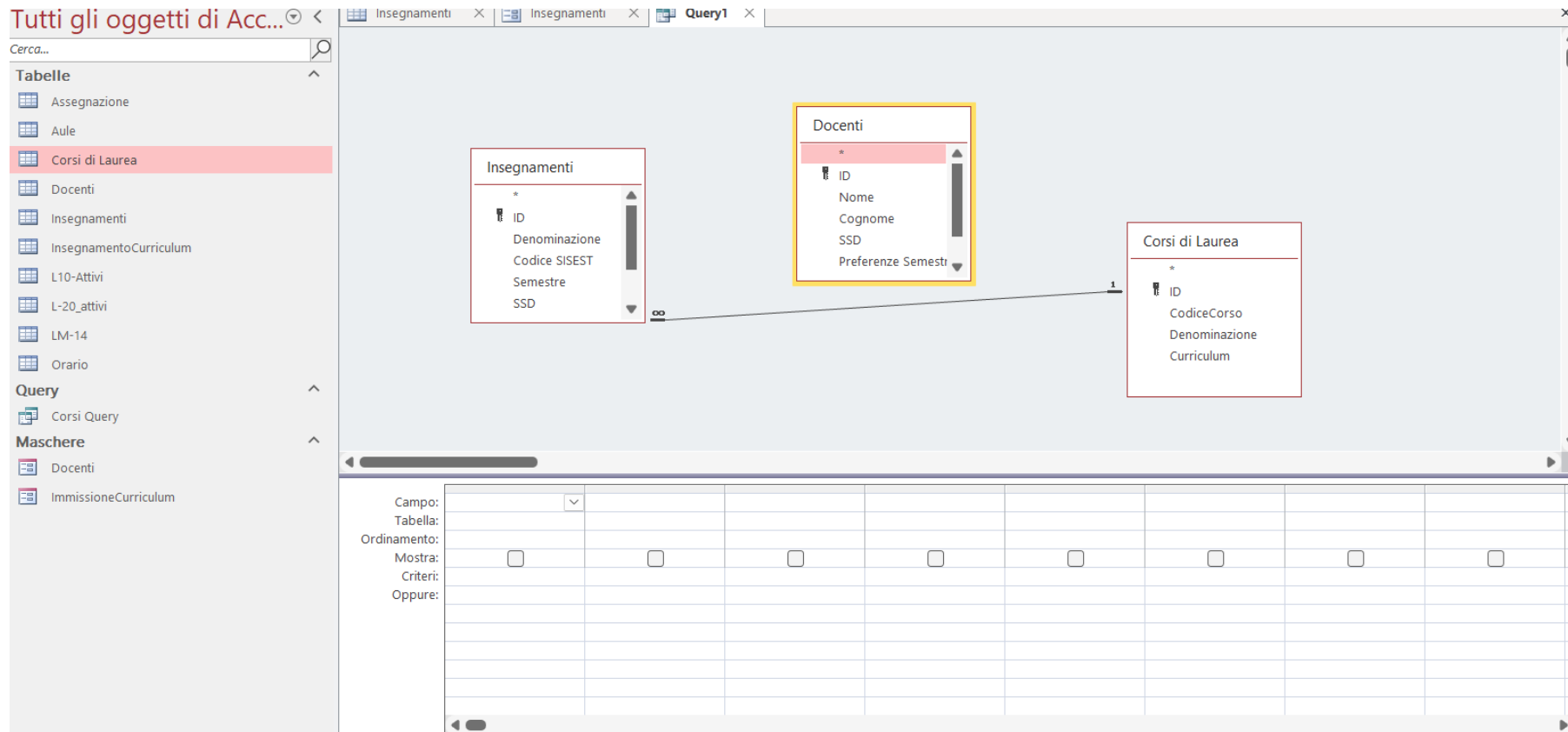
# SQL immerso in un linguaggio ad alto livello

```
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
```

# SQL immerso in un linguaggio ad hoc (Oracle PL/SQL)

```
declare Stip number;
begin
  select Stipendio into Stip
  from Impiegato
  where Matricola = '575488'
  for update of Stipendio;
  if Stip > 30 then
    update Impiegato set Stipendio = Stipendio * 1.1 where Matricola = '575488';
  else
    update Impiegato set Stipendio = Stipendio * 1.15 where Matricola = '575488';
  end if;
  commit;
exception
  when no_data_found then
    insert into Errori
    values('Non esiste la matricola specificata',sysdate);
end;
```

# Interazione non testuale



# Contenuti della Lezione

- DBMS vs file system
- Modello di rappresentazione dei dati
- Schemi ed istanze
- Architettura di un DBMS
- Indipendenza logica e fisica
- Linguaggi per basi di dati
- Attori del sistema



# Basi di dati: professionalità

- **progettisti** e realizzatori di **DBMS**
- **progettisti della base di dati** e amministratori della base di dati (DBA)
- **progettisti** e programmatori **di applicazioni**
- **utenti**
  - utenti **finali** (terminalisti): eseguono applicazioni predefinite (transazioni)
  - utenti **casuali**: eseguono operazioni non previste a priori, usando linguaggi interattivi



# Database administrator

- Persona o gruppo di persone responsabile del controllo centralizzato e della gestione del sistema, delle prestazioni, dell'affidabilità, delle autorizzazioni
- Le funzioni del DBA includono quelle di progettazione, anche se in progetti complessi ci possono essere distinzioni

# Transazioni

- Programmi che realizzano attività frequenti e predefinite, con poche eccezioni, previste a priori.
- Esempi:
  - versamento presso uno sportello bancario
  - emissione di certificato anagrafico
  - dichiarazione presso l'ufficio di stato civile
  - prenotazione aerea
- Le transazioni sono di solito realizzate con programmi in linguaggio ospite (tradizionale o ad hoc).
- **N. B.:** il termine **transazione** ha un'altra accezione, più specifica: sequenza indivisibile di operazioni (o vengono eseguite tutte o nessuna).

# Vantaggi e svantaggi dei DBMS

## **Pro**

- dati come risorsa comune, base di dati come modello della realtà
- gestione centralizzata con possibilità di standardizzazione ed “economia di scala”
- disponibilità di servizi integrati
- riduzione di ridondanze e inconsistenze
- indipendenza dei dati (favorisce lo sviluppo e la manutenzione delle applicazioni)

## **Contro**

- costo dei prodotti e della transizione verso di essi
- non scorporabilità delle funzionalità (con riduzione di efficienza)