

# CAP01\_CONCETTI\_BASE

## Table of contents

- [Concetti di base](#)
  1. [Schema, istanze e viste](#)
  2. [Modelli dei dati](#)
  3. [Linguaggi per DATABASE](#)
    1. [DDL \(Data Definition Language\)](#)
    2. [DML \(Data Manipulation Language\)](#)
  4. [Personaggi e interpreti](#)

## Concetti di base

Che cosa è una **Base di Dati (DB)**?

Un insieme organizzato di dati utilizzati per il supporto allo svolgimento delle attività di un ente, gestiti da un DBMS. Un sistema è qualcosa di complicato, non è semplicemente uno strumento ma sono tante funzioni unite.

Le basi di dati hanno alcune caratteristiche, una di queste è che sono **grandi**, parliamo di dati transazionali, dati decisionali, dati scientifici che raggiungono centinaia di Terabyte a volte.

e.g.: basi di dati bancari, a transazioni sicure

e.g.: dati scientifici, astronomici

Le basi di dati devono essere **persistenti**.

Sono **condivise**, più persone-processi condividono gli stessi dati.

Il modo di rappresentare i dati esternamente è diverso da quello interno, per poter mantenere l'**indipendenza dei dati** (**modello dei dati**).

Esistono modelli diversi di dati, ma ci occuperemo solamente del modello relazionale.

Si parla d'**indipendenza fisica** per tutti i DBMS,

si parla d'**indipendenza logica** delle tabelle di base.

Che cosa è un **Database Management System (DBMS)**?

- privacy
- affidabilità, mettendo in gioco politiche per recuperare dati per malfunzionamenti, con gestione delle **transazioni**
- efficienza, per spazio di memoria e tempo; DBMS con tante funzioni rischiano inefficienza
- efficaci, quello che deve fare viene fatto? risolve i problemi? se sì allora è un buon DBMS per i nostri usi

I **file system** estendono quelli tradizionali, fornendo più servizi in maniera integrata ("tutto o niente").

Esiste all'interno dello stesso una porzione che contiene **descrizione centralizzata dei dati**, che può essere utilizzata dai vari programmi

(catalogo o dizionario), che serve come indice.

Che cosa è una **Transazione**?

O vanno tutte a buon fine, oppure vengono riprese.

Un insieme **indivisibile** (**atomico**) corretto anche in presenza di concorrenza e con effetti definitivi.

e.g.: se al nostro conto corrente fosse addebitata una somma, ma per un errore del DBMS questa venisse

duplicata, non sarebbe una bella cosa  
Sono concorrenti, sono permanenti.

## Schema, istanze e viste

### ≡ Esempi di alcune tabelle

insegnamento	docente	aula	ora
analisi	luigi neri	n1	8:00
basi di dati	piero rossi	n2	9:45
fondamenti	luigi rossi	n1	10:00
...	...	...	...

aula	piano	dipartimento
n1	T	scienze matematiche
n2	T	informatica
...	...	...

- **schema**, intestazioni della tabella, descrivono la struttura invariante nel tempo
- **istanza**, i valori attuali che possono cambiare rapidamente, il corpo di ogni tabella
- **vista** è l'aggancio tra due tabelle che hanno una relazione.

### ≡ Un esempio di vista usando le tabelle viste sopra

insegnamento	docente	aula	ora	piano	dipartimento
analisi	luigi neri	n1	8:00	T	scienze matematiche
basi di dati	piero rossi	n2	9:45	T	informatica
...	...	...	...	...	...

## Modelli dei dati

- modelli **logici**  
sono effettivamente utilizzati dagli utenti e programmi per interfacciarsi col DBMS, modello relazionale
- modelli **concettuali**  
hanno scopo diverso, servono nelle fasi iniziali di progettazione del DBMS e servono più che altro alle persone per descrivere concetti del mondo reale, Entity-Relationship (ER)



## Linguaggi per DATABASE

- linguaggio testuale interattivo (**SQL**)
- comandi SQL in linguaggio *ospite*
- comandi SQL per interagire con linguaggio ad hoc (per grafici, istogrammi, quello che c'è dentro alla mia base di dati)
- con interfacce grafiche

## DDL (Data Definition Language)

per gli schemi

```
# creiamo una tabella nel database 'database_esempio'
USE database_esempio;

CREATE TABLE orario (
    insegnamento CHAR(20),
    docente CHAR(20),
    aula CHAR(4),
    ora CHAR(5)
);
```

## DML (Data Manipulation Language)

per le istanze

```
# inseriamo una nuova tupla nella tabella 'orario'
INSERT INTO orario
    (insegnamento, docente, aula, ora)
VALUES
    ('informatica', 'mario rossi', 'N1', '11:00');
```

## Personaggi e interpreti

- *progettisti* e realizzatori di DBMS
- *progettisti della base di dati* e *amministratori* del DB
- *progettisti* e *programmatore* di applicazioni
- *utenti*
  - finali terminalisti
  - casuali, che interagiscono con la base di dati

Le tecniche per fornire all'utente una funzione d'interesse, nella transazione, non è banale rispetto al DB. Di solito viene usato un linguaggio ad hoc per fornire funzioni.