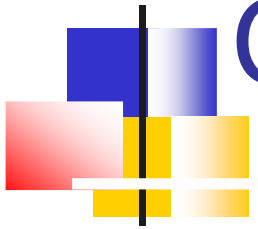
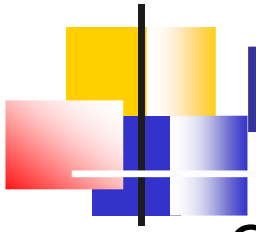


Concetti di base

Modulo B.2

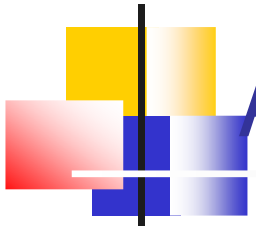


Dati B.2.1

Qualsiasi applicazione informatica gestisce ed elabora dati

- Dati interni (risultato di una elaborazione)
- I/O da e verso l'utente (interfaccia uomo-macchina)
- I/O da e verso le reti (trasmissione e trasferimento dati)
- I/O da e verso sistemi di memorizzazione permanente (banche dati)

Le applicazioni di **informatica gestionale** costituiscono la parte più rilevante dei sistemi informativi aziendali. Non hanno molte esigenze di elaborazioni, di interfaccia utente e di trasmissione, ma hanno necessità di memorizzare grandi quantità di informazioni in modo permanente. Tali dati possono essere memorizzati in **archivi** o mediante **DBMS**



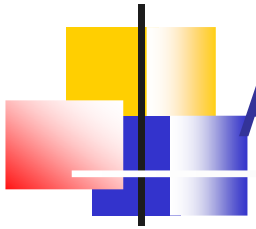
Archivi

Definizioni

Esempi

Gestione automatizzata:

- tipologia di supporto
- hardware per la memorizzazione
- Software per l'interfaccia utente
- Organizzazione



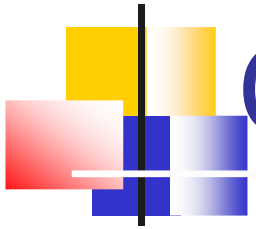
Archivi

Operazioni

Tracciato record

Record e campo

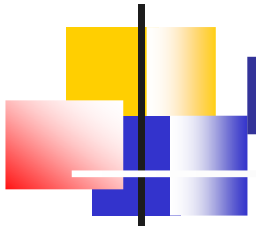
Primary key



Organizzazione degli archivi

Tipi di organizzazione:

1. Sequenziale
2. Organizzazione ad accesso diretto
3. Indexed sequential
4. Ricerca binaria sull'indice
5. Binary tree (facile la ricerca, costoso il bilanciamento)
6. Hash. Metodi per la gestione delle collisioni (collision detection)

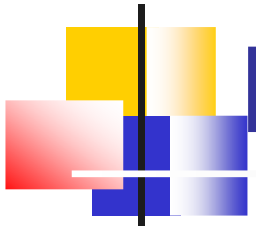


Limiti di un archivio

Gli archivi sono strumenti non integrati che possono essere utilizzati soltanto in situazioni molto semplici.

Es. i dati di una banca sono gestiti mediante due archivi (archivio clienti e archivio dei conti correnti)

In un archivio si possono aggiungere nuovi dati. Gli stessi dati possono avere formati diversi in archivi diversi. Lo stesso dato compare in posti diversi e deve essere mantenuto aggiornato (ese. L'indirizzo del correntista)

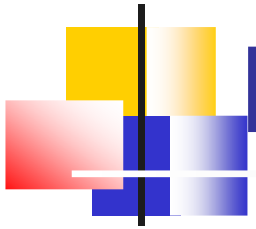


Database e DBMS

In un sistema informatico la soluzione migliore è avere una sola base dati che si interfaccia ad un DBMS

Un **database** è una collezione di dati logicamente correlati e condivisi, che ha lo scopo di soddisfare i bisogni informativi di una specifica organizzazione. I dati e la loro descrizione sono gestiti da un unico sistema chiamato DBMS

Un **dbms** è un sw che consente di costruire e gestire una base di dati, realizzandola su una memoria di massa, regolando gli accessi ai dati (pag.412)



Esempio pre DBMS

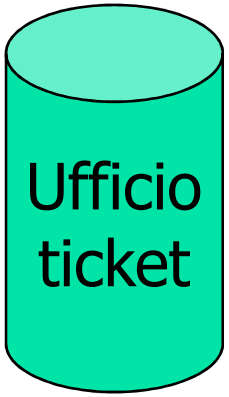
Esempio tipico dell'inconsistenza dei dati è quello di un sistema informatico di un'ipotetica azienda sanitaria che preveda due prodotti sw localizzati in due uffici diversi (uno di analisi cliniche e uno pagamento ticket)

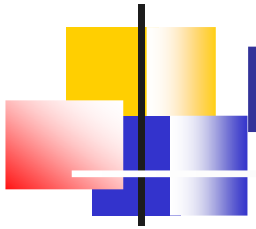
Si immagini che il sig. Rossi si rechi presso un ufficio ticket per pagare il corrispettivo di una prestazione sanitaria

Esempio pre DBMS

Applicazione

Applicazione

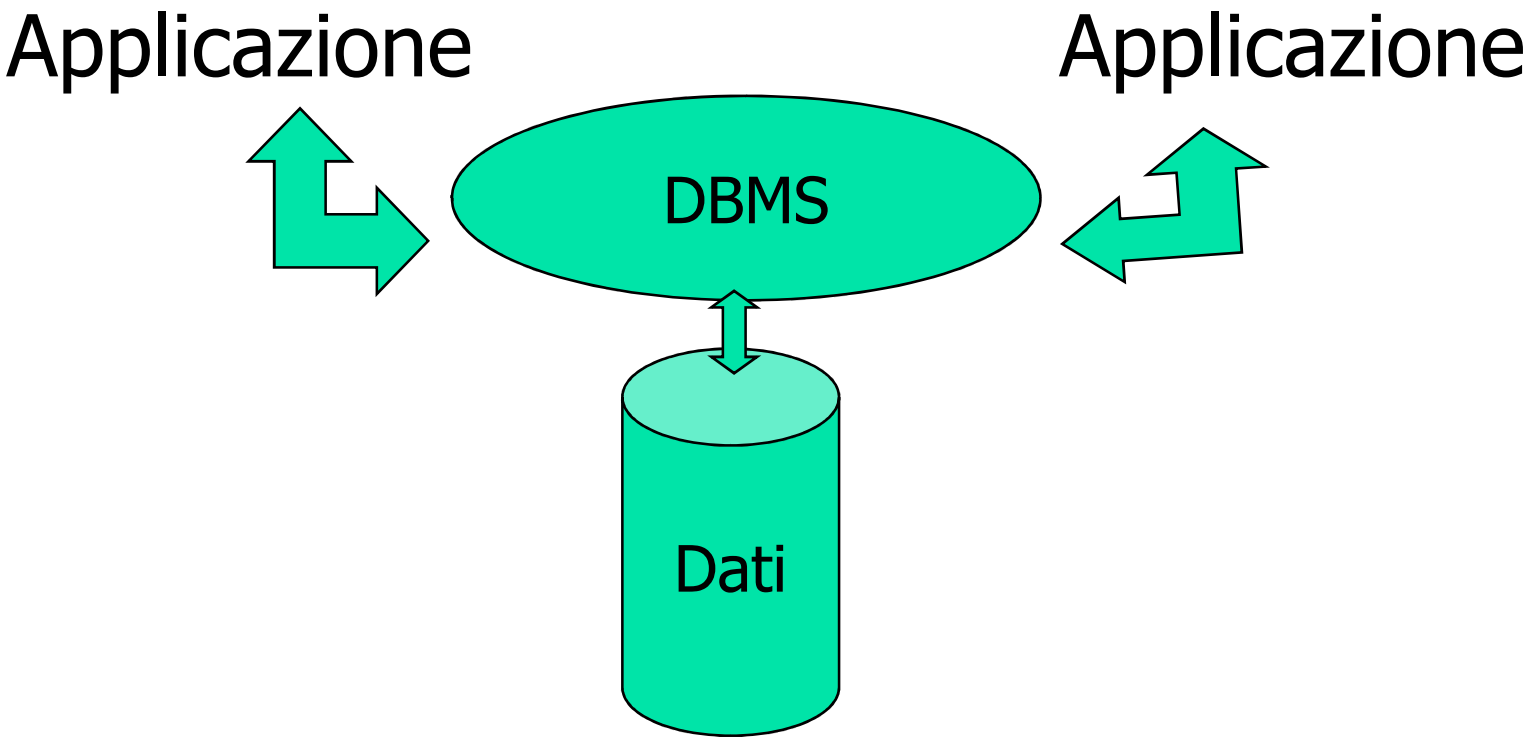


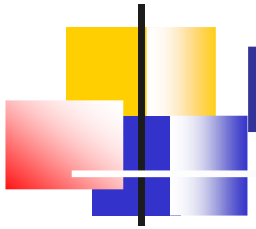


Esempio pre DBMS

Alla quarta prestazione sanitaria, e trovandosi nell'ufficio analisi cliniche, il sig. Rossi comunica la variazione di dati anagrafici perché ha cambiato abitazione. In questo caso si crea un problema di **inconsistenza** dei dati.

Esempio post DBMS



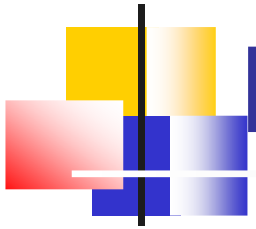


Ridondanza e integrità dei dati

Pag.413 del libro

Ridondanza: duplicazione del dato o memorizzazione di un dato che deriva dall'elaborazione di altri. La ridondanza può determinare inconsistenza?

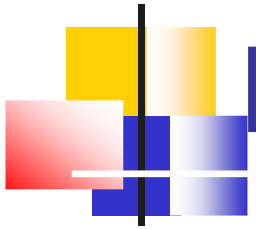
Integrità: i dati inseriti non devono essere modificati in modo errato da accessi non autorizzati in modo accidentale o provocato (esempio non posso inserire un libro in un database in una biblioteca se non ho prima inserito l'autore)



DBMS e Transazioni

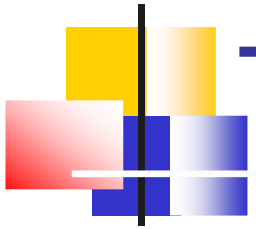
Le operazioni eseguite sui dati tramite DBMS devono essere transazioni

Una **transazione** consiste di un insiemi di operazioni di interrogazioni o modifica del DB che devono essere eseguite come se fossero un'unica operazione. Tutte le operazioni che compongono la transazione devono essere eseguite completamente e correttamente oppure non ne deve essere eseguita nessuna (pag. 421)



Esempio di transazione

Trasferimento di fondi da un conto corrente ad un altro. L'importo deve essere rimosso da un conto corrente e aggiunto ad un altro. Non è possibile che in seguito ad un malfunzionamento venga eseguita una sola delle due operazioni



Transazione ACID

Una transazione deve essere ACID

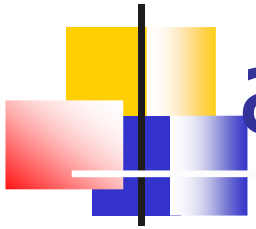
Atomicity : atomicità, la transazione è indivisibile nella sua esecuzione e la sua esecuzione deve essere o totale o nulla, non sono ammesse esecuzioni intermedie (COMMIT-ROLLBACK);

Consistency : coerenza, quando inizia una transazione il database si trova in uno stato coerente e quando la transazione termina il database deve essere in uno stato coerente, ovvero non deve violare eventuali vincoli di integrità, quindi non devono verificarsi contraddizioni (inconsistency) tra i dati archiviati nel DB

Isolation: isolamento, ogni transazione deve essere eseguita in modo isolato e indipendente dalle altre, l'eventuale fallimento di una transazione non deve interferire con altre transazioni in esecuzione

Durability: persistenza, dopo un commit work, i cambiamenti apportati non dovranno essere più persi.

Es. transazione del prelievo bancomat

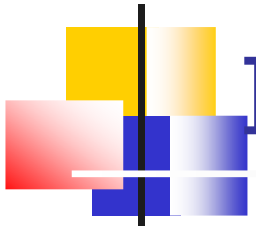


architettura

Pag.414 del libro

- Livello esterno
- Livello logico
- Livello interno

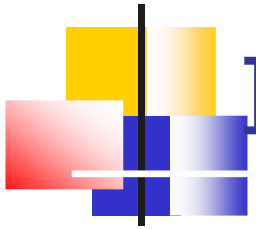
- Indipendenza logica e fisica



Integrità fisica e logica

Integrità fisica : anomalie derivate dalla lettura e scrittura dei componenti hw (libro pag.423)

Integrità logica è più complessa da identificare e gestire. Consiste nel preservare la struttura logica di una base di dati. (pag.421)

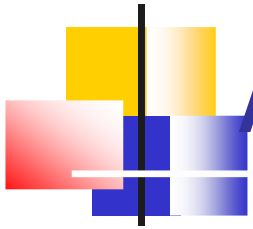


Integrità logica e transazioni

L'integrità logica si realizza con vincoli di integrità referenziale e consistenza della base di dati.

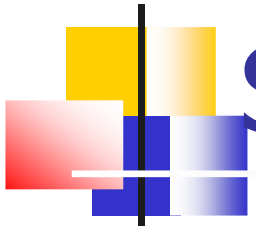
Per evitare questi problemi si ricorre al concetto di **transazione.** **Esempi di transazione: prenotazione aerea, pagamento di un bonifico, prelievo dal bancomat.**

La transazione è un insieme di operazioni che devono essere eseguite in maniera atomica (o tutto o niente).



Accesso simultaneo dei dati

In un azienda medio-piccola può esserci una sola persona che modifica i dati dei clienti. In una grossa azienda più addetti possono dover accedere una scheda di uno stesso cliente per motivi differenti (dati sul fido, o dati di marketing). La sovrapposizione crea pericolosi problemi. I DMBS utilizzano meccanismi di arbitraggio dell'accesso (**semafori**). Il primo opera e blocca, l'altro aspetta oppure rinuncia (pag.422)



Sicurezza e filtri

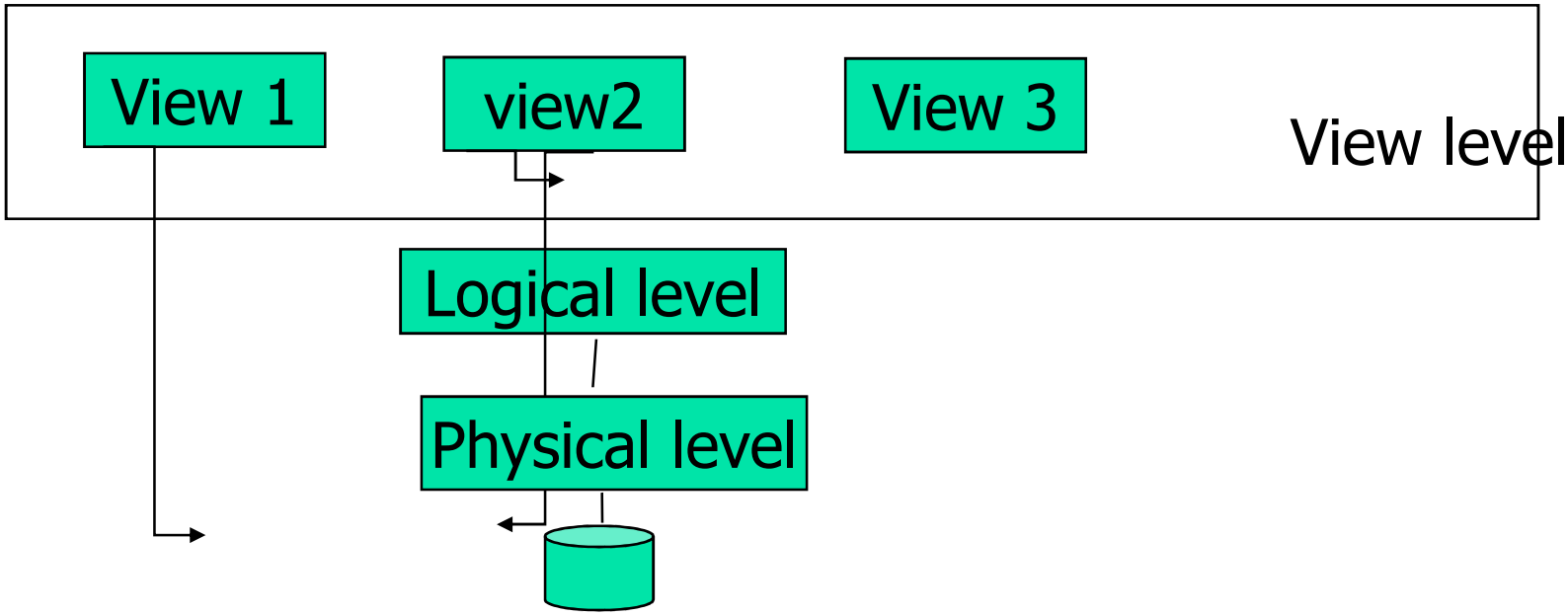
Non tutti i dati devono poter essere visibili a tutti gli utenti.

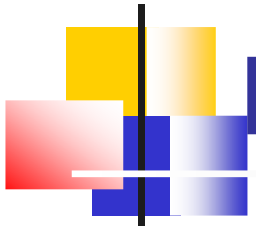
In un DBMS sono memorizzati in forma criptata, le informazioni relative agli utenti e al tipo di accesso consentito ad ognuno di essi (pag.420)

Architettura a tre livelli

I moderni DBMS utilizzano un impostazione che descrive la progettazione dei dati a tre livelli:

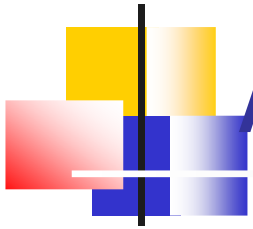
Livello esterno, logico e fisico





Livello esterno ,logico e fisico

- Il livello esterno rappresenta la visione del database da parte dell'utente (gruppo o di utenti). Lo schema esterno è diverso per ogni classe di utenza
- Il livello logico rappresentano le relazioni fra i dati senza tenere conto della memorizzazione fisica. La descrizione del livello logico avviene mediante l'uso di modelli (modello E/R, gerarchico, reticolare, **relazionale**, ad oggetti)
- Il livello fisico coincide con la rappresentazione fisica del database (es. una tabella è realizzata con archivio a indice) pag. 414 del libro



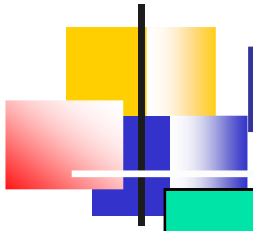
Architettura a Tre livelli

Assicura l'indipendenza dei dati. I livelli superiori non sono influenzati (entro certi limiti) dai cambiamenti che avvengono nei livelli inferiori.

(Pag.415)

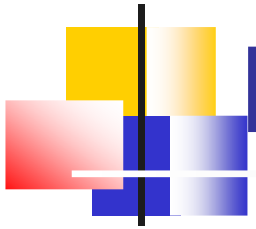
Indipendenza logica dei dati indica che uno schema esterno non viene influenzato dai cambiamenti dello schema logico

Indipendenza fisica dei dati fa riferimento alla capacità dello schema logico di non essere influenzato dai cambiamenti apportati allo schema fisico



Modelli dei dati (B.2.2)

Modellare i dati significa costruire una rappresentazione semplificata della realtà osservata o di un problema aziendale, individuandone gli elementi caratterizzanti e i legami interconnessi tra di essi

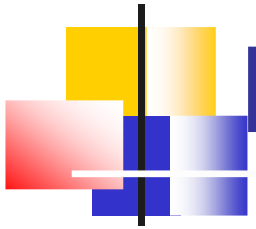


Progettazione

Pag. 424 del libro

L'uso efficace di dati organizzati presuppone un attento lavoro di progettazione iniziale. La progettazione è indipendente dal DBMS utilizzato e dai supporti fisici

Devono essere identificate le entità di interesse e le correlazioni. Questo rappresenta il modello concettuale chiamata modello E/R (entity/relationship model)

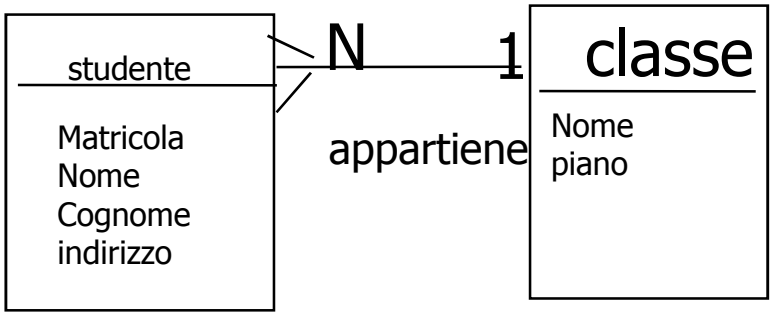


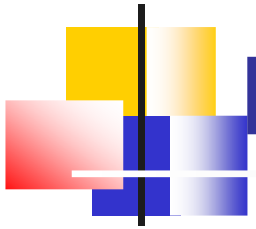
Fasi della progettazione

Pag. 425 del libro

Modelli dei dati (schema E/R)

Tra i modelli concettuali il più diffuso è il modello **E/R** (Entity/Relationship). Si identificano Entità, attributi, e le associazioni che identificano le correlazioni logiche tra entità

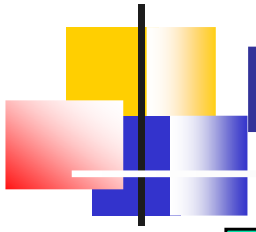




Modelli dei dati (schema logico)

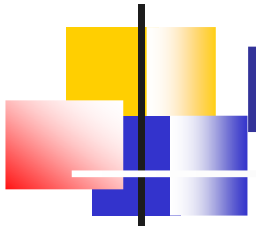
A partire dallo schema concettuale un db può essere trasformato in modello logico, cioè si analizzano le organizzazioni dei dati in modo da permettere le interrogazioni e le manipolazioni.

dagli anni 60 in poi nascevano 3 modelli logici (gerarchico, reticolare e relazionale). Il modello che è ormai diventato uno standard è il modello **relazionale**



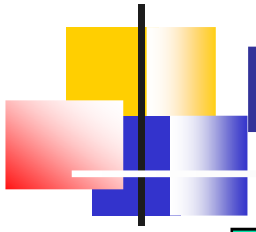
Modello gerarchico

E’ adatto per rappresentare situazioni nelle quali è possibile fornire ai dati una struttura in cui ci sono entità che stanno in alto e altre che stanno in basso secondo uno schema ad albero.



Modello reticolare

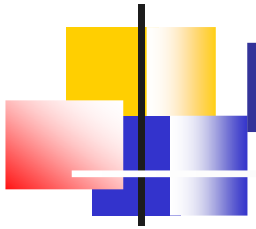
Le entità rappresentano i nodi di un grafo orientato. E' un estensione del gerarchico in quanto consente associazioni che vanno da entità che vanno dal basso verso l'alto



Modello relazionale

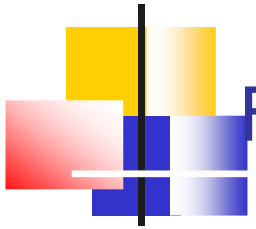
Il modello relazionale rappresenta il database come un insieme di tabelle. Esso viene considerato attualmente il modello più semplice ed efficace

Le operazioni sui DB gerarchici e reticolari sono complesse e legate alla struttura del grafo o albero. L'approccio relazionale è di tipo dichiarativo: si specifica cosa si vuol trovare non in che modo



keywords

- Architettura a 3 livelli (indipendenza logica e fisica)
- Transazione ACID
- Integrità fisica e logica
- Modello E/R
- Schema logico relazionale, gerarchico, reticolare
- View



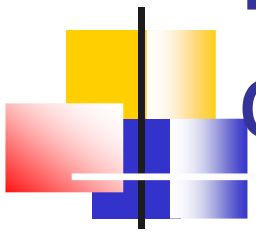
FMS (File Management System) (B.2.3)

I dati sono memorizzati nelle unità di memorizzazione. Tali unità (ad esempio i file) sono memorizzati dal sistema operativo mediante il proprio FMS .

Il FMS struttura le unità di memorizzazione in modo gerarchico mediante cartelle che contengono file o altre cartelle. Usando il FMS non è necessario conoscere come fisicamente sono memorizzati i file, basta sapere come raggiungerlo mediante il pathname.

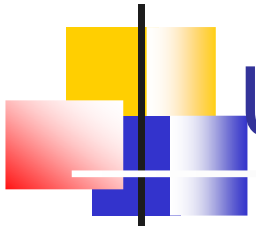
La potenza di FMS è legata alla possibilità di gestire ogni sorta di informazione, sia essa strutturata e non. E' però un limite all'efficienza, infatti alcuni DBMS non utilizzano i file tramite il FMS ma utilizzano servizi più a basso livello del S.O. scavalcando il FMS, gestendo direttamente il supporto e operando fisicamente sulle pagine. Un DBMS può gestire i dati mediante le funzioni FMS del SO oppure gestendo direttamente le unità di memorizzazione. La differenza di prestazione può essere notevole

Data dictionary o system catalogues



Un DBMS non può limitarsi a memorizzare i dati, dovrà prima di tutto memorizzare la struttura dei dati a livello logico (nomi e relazioni), sia a livello fisico (tipo di dato usato per la memorizzazione). Devono essere memorizzati anche gli indici, i dati relativi al profilo degli utenti, autorizzazione agli accessi degli utenti che accedono alla basi dati.

L'insieme di tutte queste informazioni viene chiamato **data dictionary** o **system catalogues**



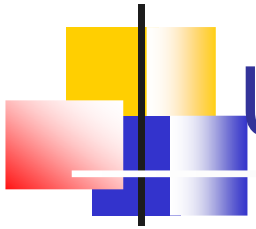
Utenti del DBMS (pag.419)

Un DBMS deve permettere la gestione di tutte le problematiche di un DB. Gi utenti sono: DBA, Programmatori e utenti finali

Un ruolo utente fondamentale è l'amministratore DBA (data base administrator) che ha la responsabilità complessiva della gestione.

Ha il compito di:

- Creare e mantenere lo schema logico (DDL)
- Definire lo schema fisico
- Definire e aggiornare i diritti di accesso (GRANT e REVOKE pag.504)
- Ripristinare la base dati in caso di malfunzionamento (Backup-restore)



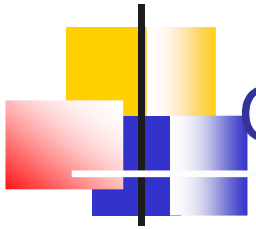
Utenti del DBMS (pag.419)

Programmatore:

Realizzano le applicazioni utilizzando il DML o particolari linguaggi di programmazione

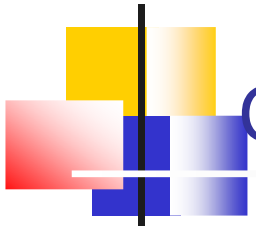
Utenti finali:

Accedono tramite le VIEW



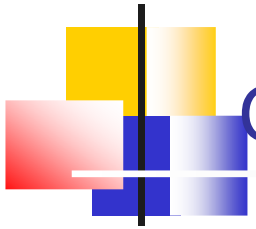
Caratteristiche di un DBMS (DDL, DML) pag.418

Permette la creazione di una nuova basedati, definendo gli archivi che la compongono, le correlazioni logiche. La creazione avviene attraverso un linguaggio ad hoc che prende il nome di **DDL (data definition language)**



Caratteristiche di un DBMS (DDL, DML)

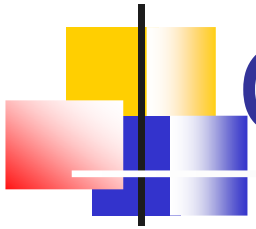
Facilita gli utenti nell'inserimento, nella cancellazione e variazione dei dati nel database sfruttando uno specifico linguaggio che prende il nome di **DML (data manipulation language)**



Caratteristiche di un DBMS (DDL, DML)

Un particolare tipo di linguaggio di manipolazione è il **QL (query language)**

Rende possibile le estrazioni di informazioni dal database interrogando i dati

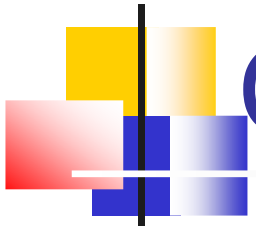


Comandi sql

SQL è un linguaggio che consente di inserire, ricercare, aggiornare, cancellare i dati di un database di tipo relazionale

SQL è un linguaggio di tipo non procedurale.

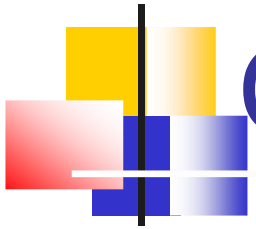
Può essere utilizzato in modo interattivo (ciò lo eseguo ed ottengo il risultato) sia embedded cioè all'interno di altri linguaggi di programmazione



Comandi DDL (pag.482)

```
CREATE TABLE Giocatori
(Cognome CHAR(20) NOT NULL,
  Nome CHAR(35),
  Sesso CHAR(1) );

CREATE UNIQUE INDEX indice_cognome
ON Giocatori (Cognome);
```



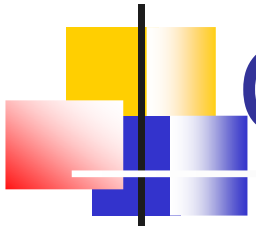
Comandi DDL (pag.482)

```
ALTER TABLE Giocatori
ADD Punti INT;

ALTER TABLE Giocatori
DROP Punti INT;

DROP TABLE Giocatori;

DROP INDEX indice_cognome;
```



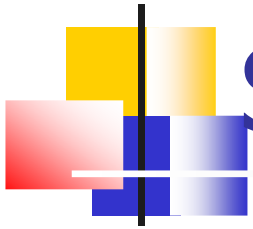
Comandi DML (pag.500)

SELECT……

INSERT INTO Giocatori
VALUES (‘Rossi’, ‘Mario’, ‘M’;

UPDATE Giocatori
SET Sesso=‘F’ WHERE Sesso=‘M’;

DELETE FROM Giocatori WHERE Cognome=‘Rossi’;



Sicurezza e integrità dei dati

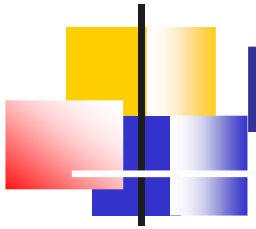
In un DBMS la sicurezza gioca un ruolo importantissimo:

acronimo CIA indica :

Confidentiality (riservatezza)

Integrity (integrità)

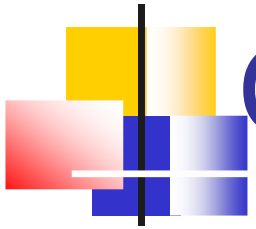
Availability (disponibilità)



Minacce alla sicurezza

Possono arrivare da diverse parti:

1. Fattori umani (disattenzioni o vere e proprie manomissioni di dati)
2. Fattori fisici (guasti alle apparecchiature, furti,..).
3. Fattori legati ai sistemi operativi su cui un dbms si appoggia (bachi sw, errori nella gestione della sicurezza del SO,..)



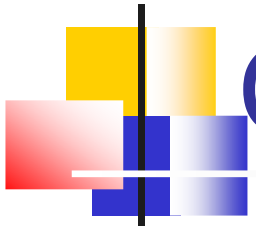
Contromisure

Per garantire la **confidentiality**:

- assegnare permessi limitando gli accessi
- effettuare copie di backup

Per garantire **l'integrity**:

- limitare gli accessi, attenta gestione delle transazioni



Contromisure

Per garantire **l'availability**:

- corretto dimensionamento dell'HW
- politiche di memorizzazione su dischi ridondanti (dischi RAID)
- on line logging (tracciamento delle operazioni effettuate)