

# Gestore delle strutture di memorizzazione / P - 03

Tempo di accesso e trasferimento:

$$t = \text{latenza} + \frac{\text{dimensione dati da trasferire}}{\text{velocità di trasferimento}}$$

tempo x occorso  
el 1° byte di interesse

(→ due memorie: da secondario a primaria)

Tale somma concretizza le prestazioni di una memoria

- Tempo di accesso alle memorie
- Velocità a cui gli I/O avvengono

- I dati: di dati risiedono su memoria non volatile (dischi)
- I dati per essere elaborati devono essere trasferiti da memoria non volatile a cerchi
- I dati vengono spostati a blocchi (non bit x volte)
  - ↓
  - attraverso un buffer al quale si può lavorare sui dati

## Gestore dei file

- L'informazione è memorizzata in ordini di piccole superfici detti **tracce**
  - Ogni traccia può essere divisa in più unità (minima ~ 512b) dette **settori** dopo che la lettura si è posta sulla traccia
  - Le tracce con stesso diametro sulle diverse superfici sono dette **cilindri**; mi basta muovere una volta la testina (movimento meccanico più costoso) sapendo che tutte le informazioni di mio interesse si trovano sullo stesso cilindro  
→ dati recuperabili più velocemente se sullo stesso cilindro
  - Tempo di latenza nel caso peggiore  $\approx 28 \text{ ms}$
- La dimensione del blocco (detto pagina quando caricato in memoria centrale) dipende: 4-64 Kb → più piccolo è il blocco più operazioni di I/O (meglio tempo) → motivo dell'op  
→ più grande invece pesa sul caricamento e prende scorrere più spazio in memoria (di per sé i dati sono memorizzati frammentati e in modo casuale)

Prestazioni della memoria dipendono dal tempo di trasferimento e dalla dimensione dei blocchi.

Il gestore delle strutture di memorizzazione deve cercare di ridurre il tempo di latenza ( $\gg$  tempo di trasferimento dati)

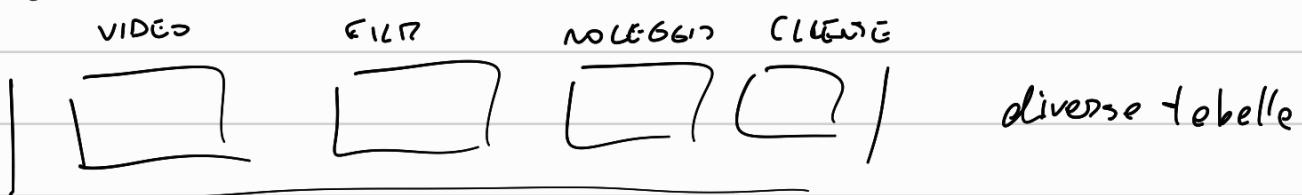
## DB fisic

Il DB consiste di un insieme di file visti come collezione di pagine (blocki) di dimensione fissa.

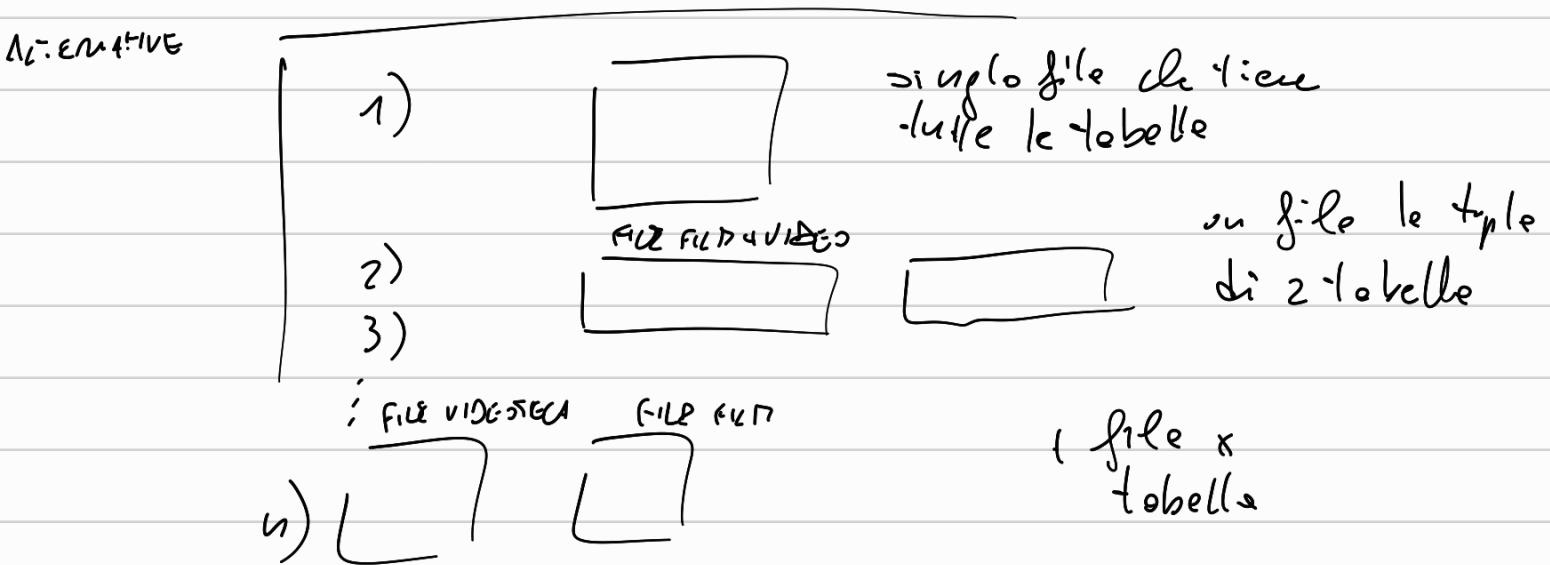
ogni prefine memorizza più record (tuple)

A loro volta i vered consistono di più campi: di lunghezza  
fissa che rappresentano gli attributi

A livello logico



## A livello fisico



## Relazione super e record

Significato di una dupla corrisponde a un record

I record sono salvati in file ↪  
lunghezza fissa → fissa (inizio  
di record)  
lunghezza variabile

In file puo' avere lunghezze variabili per varie ragioni:

- campi di dimensione variabile (es. "titolo" c'è una stringa di dim var  
→ salvo negli spazi vuoti o separatori)
- esprimere campi optional
- descrivere fileeterogenei (piu' tuple di relazioni diverse)

Per la lunghezza variabile si puo' usare:

- l'idea di tutti e soli i campi di testo sono variabili
- i campi di fissa sono posti prima di quelli a lunghezza variabile
- un "prefix pointer" che ripete l'indirizzo del primo byte del campo

Ogni record contiene un **header** che specifica

- identificatore della relazione specificata
- " " del DB
- timestamp che indica quando il record e' inserito/modificato

Soltanente un record e' contenuto interamente in una pagina.

Il **page header** contiene info sui record (timestamp etc...)

Un record si sente di memorizzare a casello di pagine diverse:

→ occedere alle tuple dovei spostarmi su pagine diverse (disperdimento)

Sul disco i blocchi sono memorizzati con **allocazione contigua**,

ovvero sulle stesse tracce o sullo stesso cilindro ("densi")

→ maggiore efficienza in lettura

→ grossi problemi: se devo modificare (es. aggiungere film)  
lo spazio delle celle successive e' occupato e devo spostare  
tutti i blocchi relativi alla stessa relazione (es. film))

Ottimale utilizzo con allocazione concatenata: ogni blocco ha un punto di origine chiamato "contiguo"

→ grande spostamento delle testine?

No se utilizza un **BUCKET** ovvero memorizza i blocchi in modo "vicino" ovvero sullo stesso cilindro per gruppi di record collegati

I file da contengono tutti i dati (rappresentano tutte le tuple del DB) e rappresentano l'organizzazione primaria dei dati

→ il modo con cui vengono salvati determina l'efficienza delle operazioni e sull'occupazione di memoria

Le caratteristiche dell'organizzazione primaria (es. file part dal blocco n° 3 etc...) sono salvate in **CATALOGHI DI SISTEMA** che dipendono da DBRS e DBPS.

Contengono info su

→ n° tuple

→ n° blocchi nel file e altre statistiche

Anda l'ordine con cui memorizza i record incidenza sulla prestazione.

→ es se ho un ordine passo entroso di leggere blocchi "intili" rispetto alle richieste fatte (es. valggi fra due file, se conoscendo la posizione del blocco con la prima file c'ha un ordine capovolto)

