

# Gli archivi di dati

---

DI ROBERTA MOLINARI

# Gli archivi

I **dati** sono una rappresentazione dei fatti.  
L'**informazione** è l'incremento di conoscenza  
che può essere acquisita dai dati.

---

Gli **archivi** servono per conservare i dati in modo permanente, per poter essere reperiti e utilizzati in seguito.

Gli archivi sono un insieme di dati i cui elementi hanno le seguenti caratteristiche:

- sono legati tra loro da un *nesso logico* (si riferiscono ad uno stesso argomento Es. rubrica telefonica)
- sono rappresentati secondo un certo *formato* per permetterne l'interpretazione (Es. nominativo, indirizzo, tel)
- sono registrati in modo permanente su un certo *supporto* su cui è possibile leggere e scrivere (Es. la rubrica cartacea)
- sono *organizzati* per permetterne la consultazione (Es. ci sono le etichette delle lettere)

# Gli archivi i dati

---

Ogni dato numerico o alfabetico, viene memorizzato tramite una successione di byte

Esempio:        stringa "AB"

ASCII("A") = 65 in binario 01000001

ASCII("B") = 66 in binario 01000010

In MM sarà memorizzata la successione di byte

**01000001 01000010**

Esempio:        valore numerico 16706

16706 in binario 0100000101000010

In MM sarà memorizzata la successione di byte

**01000001 01000010**

# Gli archivi

## Record logici

---

I dati, in generale, sono raggruppati in unità logiche ognuna riferita ad un singolo soggetto della “realtà” memorizzata.

Ogni unità è un **record logico**, che a sua volta è suddiviso in **campi**, i cui valori caratterizzano il soggetto. L'elenco dei campi e il relativo tipo costituiscono il **tracciato record**.

Es. di tracciato record per la rubrica telefonica

| Cognome-Nome | Indirizzo    | Numero Telefono |
|--------------|--------------|-----------------|
| Alfabetico   | Alfanumerico | Numerico        |

Es. di record

|             |            |         |
|-------------|------------|---------|
| Rossi Mario | Via Roma 1 | 5556346 |
|-------------|------------|---------|

# Gli archivi

## Record logici

**Campo:** spazio riservato per memorizzare il dato relativo ad un'informazione di senso compiuto (es. indirizzo). Singolo dato

**Record:** insieme di campi correlati tra loro formanti un'unità informativa più ampia (es. persona).

Insieme di dati di uno stesso soggetto

| Nome          | Indirizzo       | Telefono   |
|---------------|-----------------|------------|
| Mario Rossi   | Via Roma 1      | 0171-66666 |
| Bianchi Luisa | P.zza Italia 14 | 011-999999 |
| Verdi Ugo     | Via Po 12       | 011-444444 |

**Archivio:** insieme di record omogenei (es. rubrica).

Insieme di dati di un archivio di più soggetti

# Gli archivi

## Operazioni

---

- **Creazione dell'archivio:** si definisce il supporto, il tracciato record, il nome e l'organizzazione
- **Manipolazione dei dati:**
  - *Inserimento*
  - *Modifica o aggiornamento*
  - *Cancellazione*
- **Interrogazione o consultazione dei dati:** reperimento di informazioni
- **Distruzione**

# Gli archivi elettronici

## Memoria di Massa

---

- ▶ Quando l'archivio è memorizzato su un computer, si parla di **archivi elettronici**. A tale scopo si utilizzano le **Memorie di Massa**: memorie **permanenti** (non volatili), contenenti grosse quantità di dati.
- ▶ Le **unità disco** o **drive** permettono la lettura/scrittura dei dati e il loro trasferimento dalle memorie di massa alla memoria RAM e viceversa.
- ▶ Esistono diversi tipi di memorie:
  - Nastri
  - Disco Rigido (Hard Disk)
  - CD-ROM CD-R CD-RW DVD
  - Pen Drive o Key USB
  - Schede SD (Secure Digital)
  - Dischi allo stato solido SSD (Solid State Drive)



# Memorie di massa

## Hard Disk

---





# Memorie di massa

## Hard Disk

---

### I **dischi rigidi HDD** (Hard Disk Drive):

- sono dotati di una serie di **piatti rotanti** sui quali scorrono le **testine con braccio meccanico** con la capacità di leggere, scrivere e sovrascrivere i dati.
- I piatti, da 3.5 o 2.5 pollici, sono di solito prodotti in vetro o alluminio e ricoperti da un sottile strato di metallo, sui quali si trovano suddivise delle micro-celle che possono *essere magnetizzate o smagnetizzate* in fase di registrazione dei dati.

# Memorie di Hard Disk

**Cluster:** insieme di sector contigui, (da 1 a 128). In windows corrisponde ai blocchi fisici. Unità minima trattabile dal SO. È > di un sector per limitare la frammentazione

I dischi sono le periferiche più utilizzate

Sono suddivisi in:

- tracce [1000-5000]
- settori [64-200]

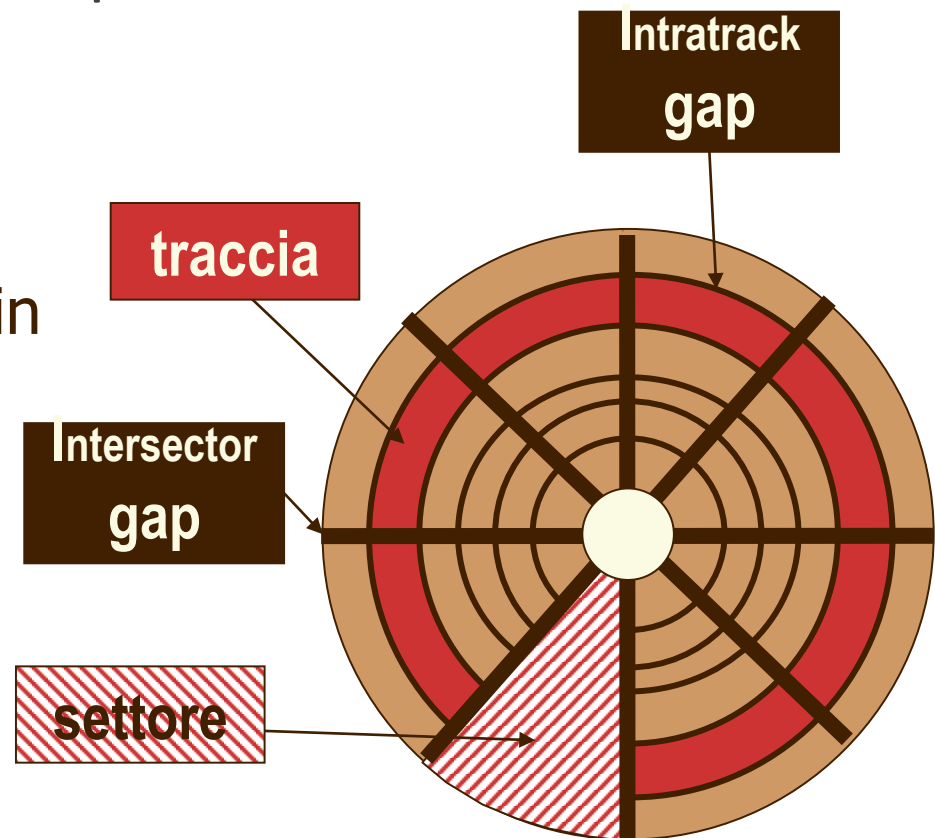
I dati sono memorizzati in

**sector** o **record** o

**blocchi** tutti di uguale capacità (512B - 4KB)

individuabili tramite:

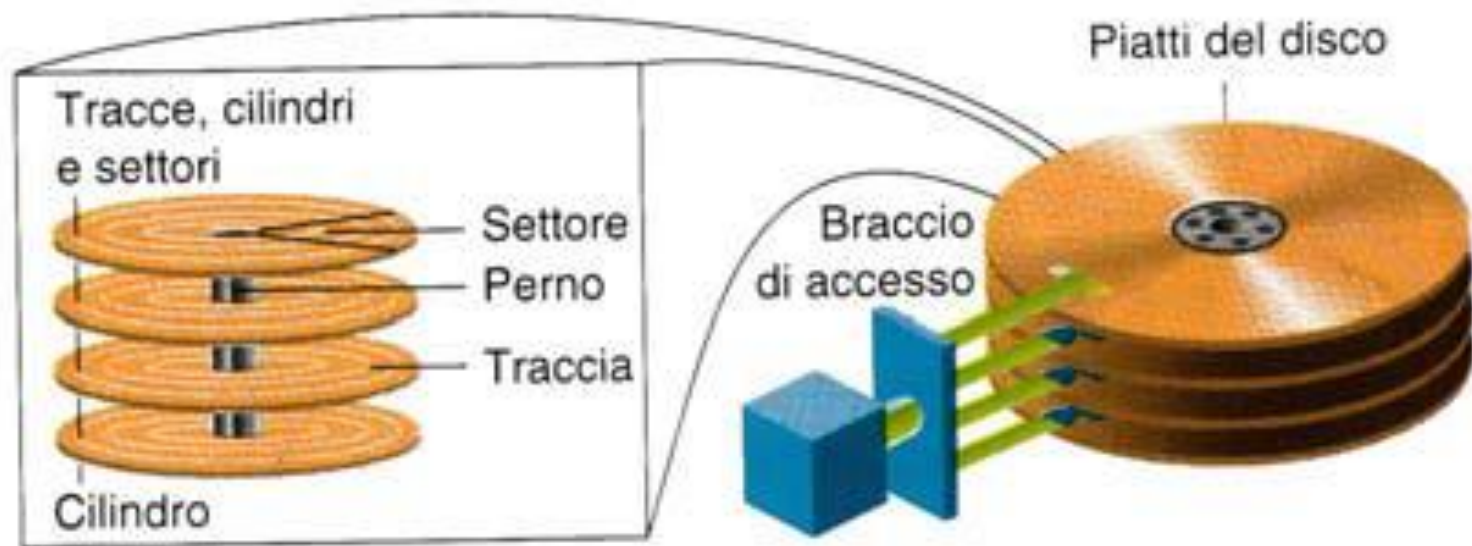
- n° facciata
- n° traccia
- n° settore



# Memorie di massa

## Hard Disk

Un disco rigido è formato da una pila di dischi (*diskpack*). L'accesso ai dati è fornito da un gruppo solidale di bracci con una testina sia per la facciata superiore che inferiore. Le tracce corrispondenti su più facciate formano un **cilindro**.



*La struttura di un disco fisso*

# Memorie di massa

## Hard Disk

---

L'accesso ai dati avviene in modo **semidiretto** o **semicasuale**, “diretto” alla traccia e alla facciata, sequenziale al settore

Il tempo di accesso dipende da:

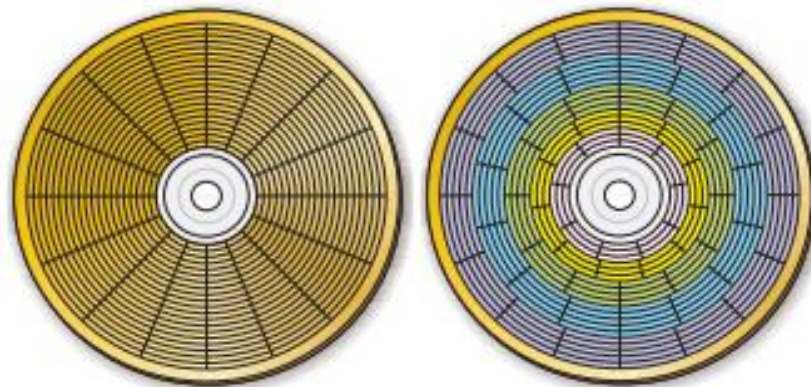
- **Tempo di seek:** per accedere alla traccia
- **Tempo di latency:** per accedere al settore
- **Tempo di trasferimento:** per trasferire fisicamente i dati da e verso la MC

# Memorie di massa

## Hard Disk

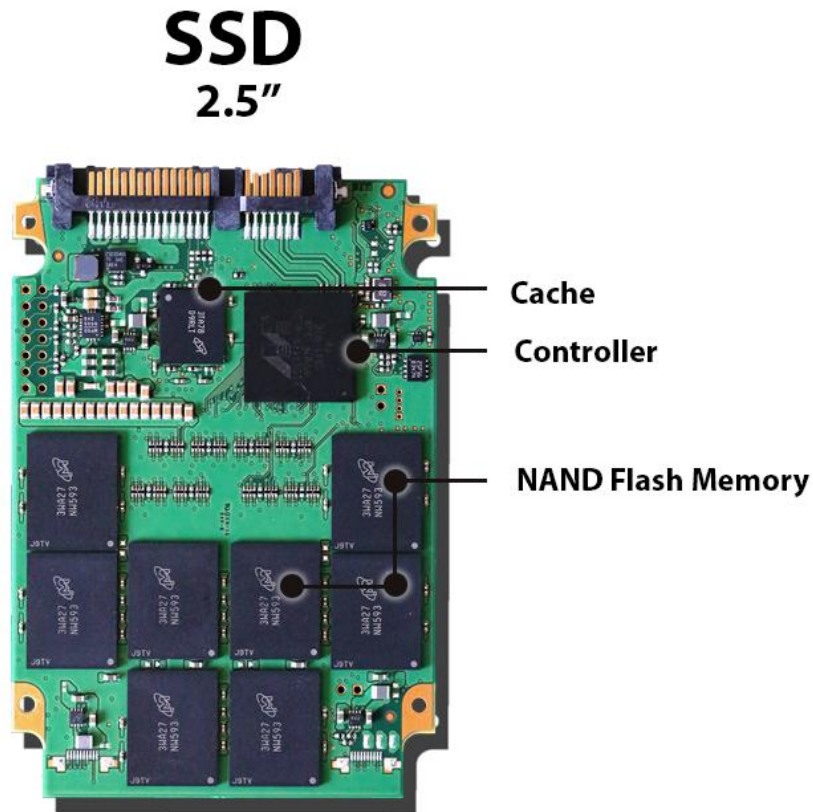
---

Un modo di aumentare la capacità dei dischi si basa su una tecnica di registrazione dei dati su disco detta **multiple zone recording**, o semplicemente zone recording, **registrazione a zone**. La registrazione a zone viene attuata aumentando il numero di settori per traccia nel passare dalle tracce interne alle tracce esterne.



# Memorie di massa SSD

---



# Memorie di massa SSD

---

- In un **disco allo stato solido SSD** (Solid State Disk) i bit dei dati sono salvati attraverso dei transistor MOSFet (mantengono una carica elettrica e non una carica magnetica) (Memorie Flash)
- È fisicamente organizzato in **pagine** (di solito da 4kByte), identificate da uno o più valori numerici. La pagina rappresenta *l'insieme minimo di dati che è possibile leggere/scrivere in un'unica operazione*. Le pagine di un SSD possono essere riscritte un numero limitato di volte (fra 3000 e 100000 riscritture per ogni cella).
- I driver degli SSD implementano opportuni algoritmi per garantire un utilizzo uniforme delle celle (non vengono scritte sempre le stesse) che elimina la frammentazione.

# Memorie di massa SSD

---

I dischi SSD sono composti da quattro parti cardine:

- il **controller** o **microprocessore**: ha un certo numero di core, organizza le operazioni di memoria di massa ed è governato da un firmware;
- la **memoria cache**: serve per registrare in modo temporaneo le informazioni da e verso la memoria;
- il **supercondensatore**: simile a una batteria, permette di terminare la scrittura anche in assenza di alimentazione;
- l'**interfaccia**: permette il collegamento, sia dati che di alimentazione, con la scheda madre con collegamenti SATA o PCI EXPRESS.



# Memorie di massa

## File system

---

- Il **file system** vede HDD e SSD come una serie di numerose locazioni di memoria (pagine o settori di traccia) identificati da un *indirizzo*, quindi per il SO la gestione dei due dispositivi è sostanzialmente identica.
- I file sono suddivisi in **blocchi logici** e memorizzati in una sequenza non contigua di **blocchi/pagine fisici** ciascuno identificato dalle proprie coordinate di superficie, traccia, settore (per gli HDD) o dall'identificatore di pagina (per gli SSD).

# Memorie di massa

## blocco – fattore di blocco

---

Le MM sono suddivise in successioni di byte

Ad ogni operazione di I/O vengono trasferiti da o per la MM blocchi di byte indivisibili che formano il **record fisico**  
(possono corrispondere ad 1 o più sector contigui della stessa traccia)

**Record fisico:** minima quantità di dati trattabile in una singola operazione di I/O. Non esiste una suddivisione in campi.

In generale un record fisico non coincide con il record logico. Il loro rapporto determina il **fattore di blocco**

**Fattore di blocco:** n° di record logici che compongono un blocco

=1 record **sbloccati**, blocco coincide con il record logico

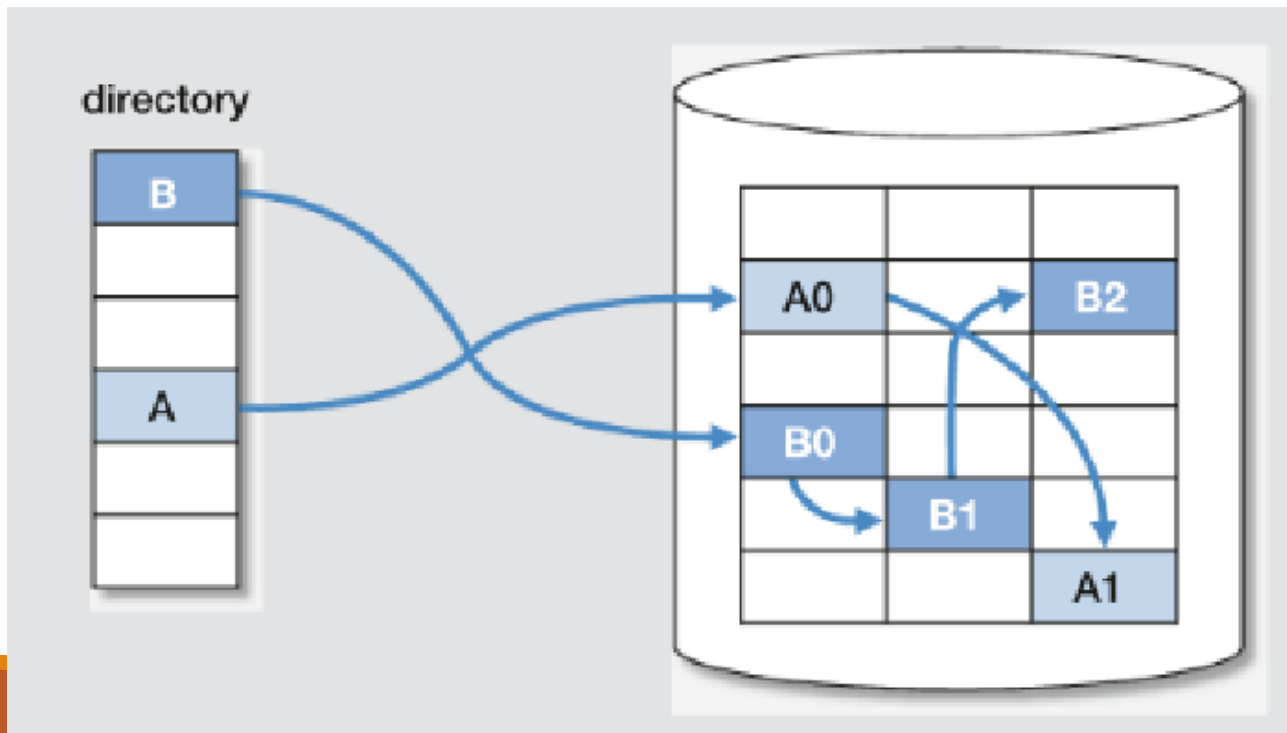
>1 record **bloccati**, se in un blocco ci sono più record logici

<1 record **multiblocco**, se ci vogliono più blocchi per un record logico

# Memorie di massa

## I file

I file sono memorizzati su più blocchi fisici. Ogni blocco fisico contiene memorizzate, oltre ai dati del file, le coordinate del blocco successivo, o in alternativa l'indicazione che si tratta dell'ultimo blocco del file.



# Organizzazione fisica MM

## lunghezza record: fissa

---

**Record a lunghezza fissa:** contengono lo stesso numero di campi nello stesso ordine e ciascun con lunghezza prefissata costante; il dato da inserire subirà un "troncamento" se eccede la lunghezza o sarà "pareggiata" con riempimenti non significativi (blank). Si ha il vantaggio di poter effettuare inserimenti e cancellazioni di record non solo in coda (append), ma anche in punti intermedi in quanto si calcola l'indirizzo del N-simo record con la formula:

$$\text{IndRecN} = \text{IndBase} + \text{LungRecLog} * (\text{N} - 1)$$

|           |           |           |           |        |
|-----------|-----------|-----------|-----------|--------|
| Canguro-- | Squalo--- | Gatto---- | Canarino- | ...    |
| 9 car.    | 9 car.    | 9 car.    | 9 car.    | 9 car. |

# Organizzazione fisica MM

## lunghezza record: variabile

---

**Record a lunghezza variabile:** possono contenere campi a lunghezza variabile ed in numero variabile; il tracciato record è definito di volta in volta dall'utente in base alle operazioni (immissione/modifica dati) ed i campi sono lunghi tanto quanto il dato in essi memorizzato, se questo è assente il campo è nullo. Insieme ai dati sono memorizzate informazioni gestionali (identificatori, marker, specificatori di lunghezza). Si ha meno spreco di memoria, ma i record si possono aggiungere solo in coda e una cancellazione comporta la rielaborazione del file.

|   |         |   |        |   |       |   |          |     |
|---|---------|---|--------|---|-------|---|----------|-----|
| 7 | Canguro | 6 | Squalo | 5 | Gatto | 8 | Canarino | ... |
|---|---------|---|--------|---|-------|---|----------|-----|

# Organizzazione fisica MM

## lunghezza record: indefinita

---

**Record a lunghezza indefinita:** la lunghezza è variabile, ma non stabilita a priori. L'inizio e/o la fine del record sono individuati tramite un carattere di controllo memorizzato con il record.

|          |         |        |           |     |
|----------|---------|--------|-----------|-----|
| Canguro@ | Squalo@ | Gatto@ | Canarino@ | ... |
|----------|---------|--------|-----------|-----|

# Memorie di massa

## Le directory

---

Le directory risiedono in MM.

Una directory è un file particolare che contiene una tabella contenente le informazioni per il reperimento di ogni file o spazio libero presente sul supporto di massa su cui risiedono:

| Indice | Nome File | Lunghezza Record | N record | Ind. 1° blocco | Accesso |
|--------|-----------|------------------|----------|----------------|---------|
| 1      | Servizio  |                  |          |                |         |
| 2      | Servizio  |                  |          |                |         |
| 3      | CLIENTI   | 400              | 20       | 46             | R       |
| 4      | Libero    | 200              | 50       | 86             |         |
| 5      | ELENCO    | 100              | 100      | 136            | R/W     |
| 6      | Libero    | 200              | 30       | 186            |         |
| ...    |           |                  |          |                |         |

# Le directory tabella dei file aperti

---

La directory è anch'essa un file da dover caricare, del tutto o in parte, in MC.

Per velocizzare le ricerche dei riferimenti dei file su cui si ritiene di dover agire frequentemente, si utilizza una tabella presente permanentemente in MC, la **tabella dei file aperti**.

- Ogni istruzione di OPEN aggiunge una riga nella tabella
- Ogni istruzione di CLOSE libera una riga



# Le directory

## struttura delle directory

---

Le directory sono organizzate in varie strutture logiche che devono permettere in modo efficace le seguenti operazioni:

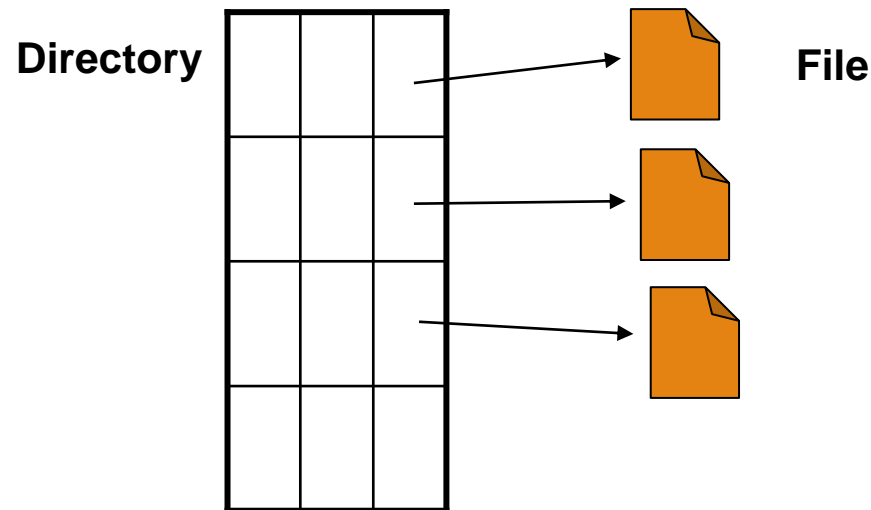
- *Aggiunta* di un nuovo file
- *Eliminazione* di un file
- *Elencazione* dei file presenti sulla directory
- *Ricerca* di un file

# Le directory

struttura delle directory a livello singolo

---

Tutti i file della MM sono memorizzati in un'unica directory



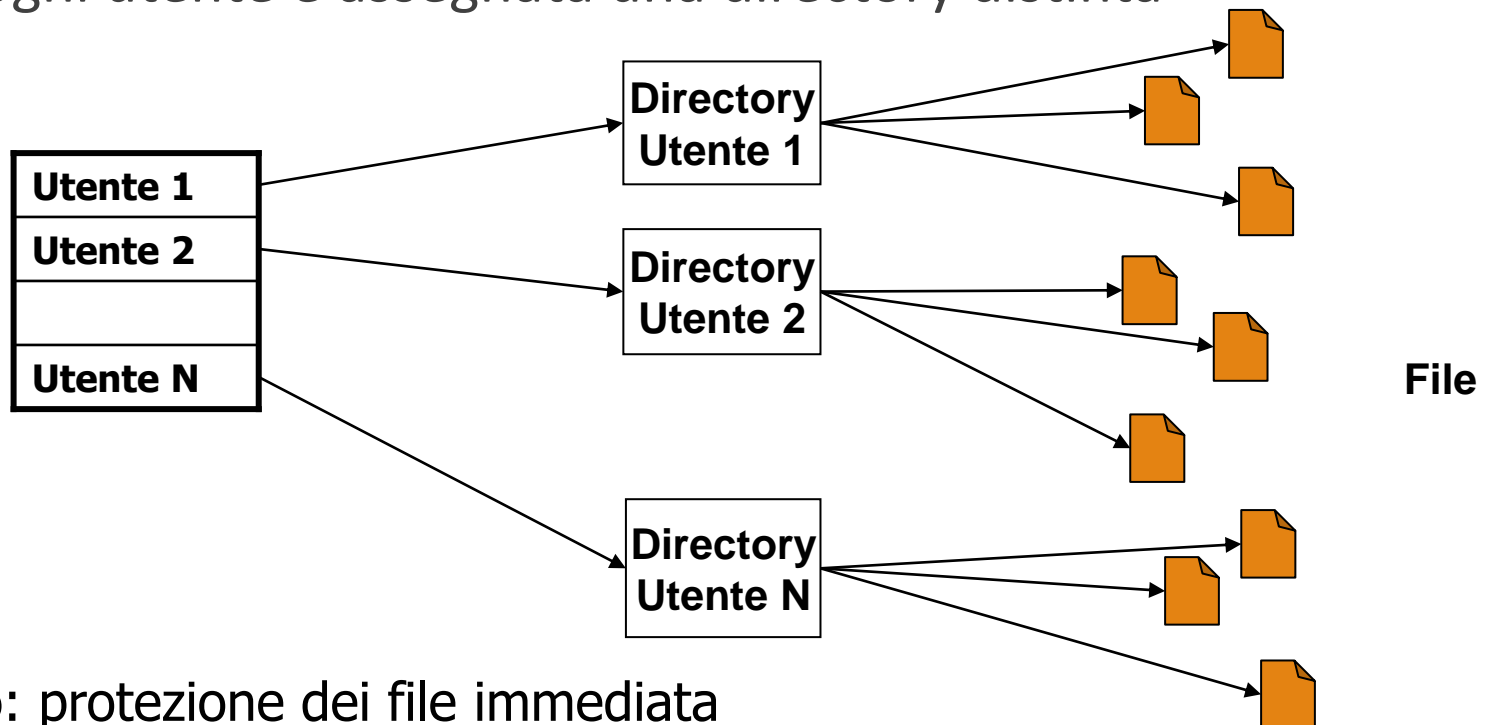
**Pro:** molto semplice da implementare

**Contro:** problemi se il n° di file cresce, o se la MM è condivisa da più utenti. Non ci possono essere 2 file con lo stesso nome

# Le directory

## struttura delle directory a due livelli

Ad ogni utente è assegnata una directory distinta



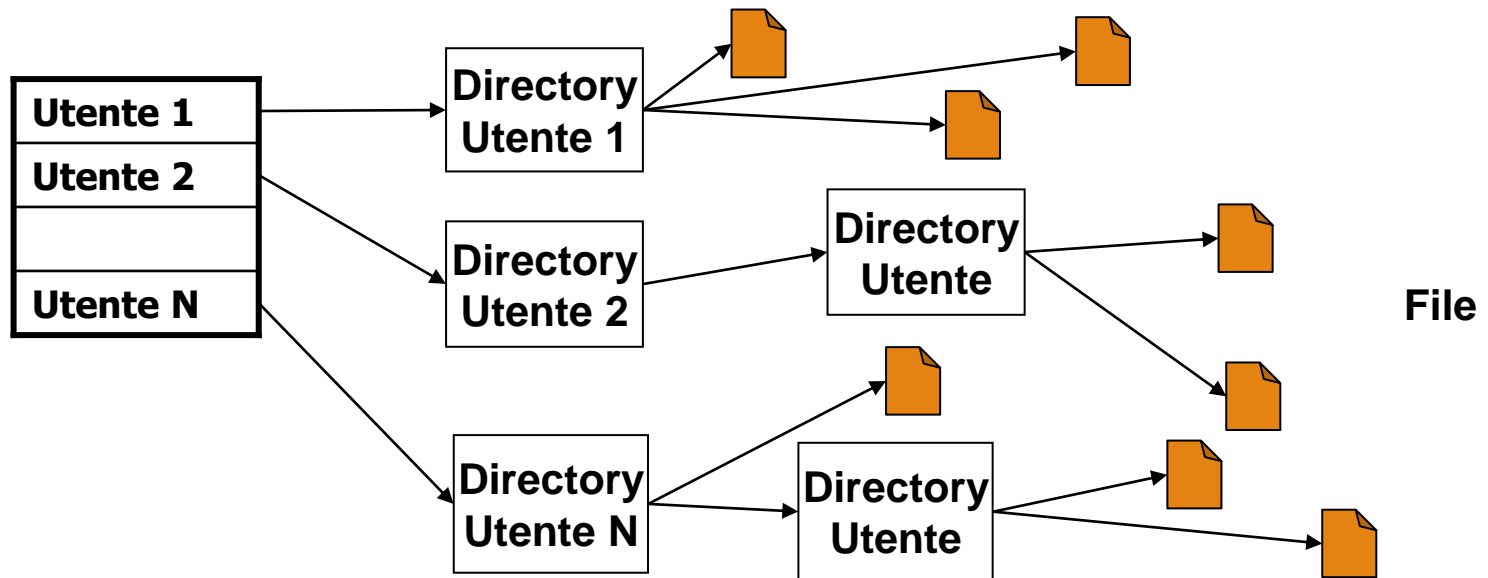
**Pro:** protezione dei file immediata

**Contro:** non si possono condividere dei file, ma solo duplicarli, compresi quelli di sistema

# Le directory

## struttura delle directory ad albero

Esiste una directory principale (root) da cui derivano tutte le altre. I file hanno un *Pathname*



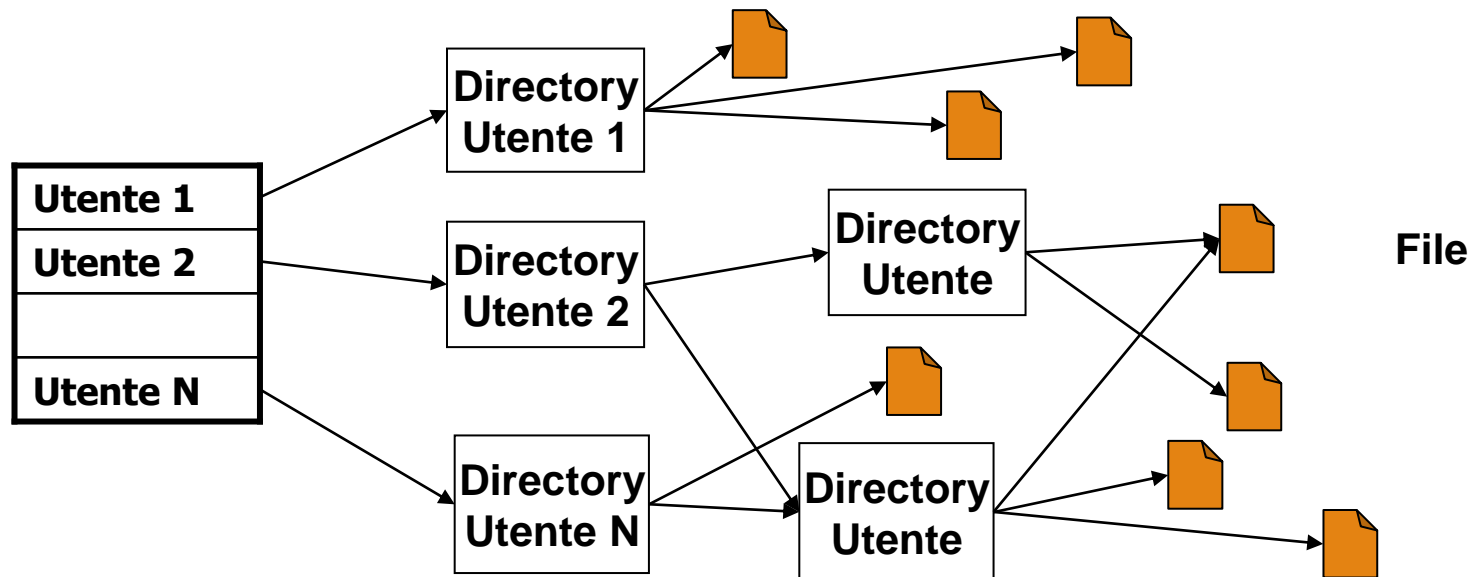
**Pro:** si può accedere ai file di altri tramite il pathname. Si possono avere file con nomi uguali

**Contro:** per avere il file di un altro sotto la mia directory, devo copiarlo

# Le directory

## struttura delle directory a grafo

È una struttura ad albero con la possibilità di accedere ad uno stesso file tramite pathname diversi



**Pro:** condivisione totale dei file

**Contro:** struttura complessa, protezione dei file più complessa

# Organizzazione logica MM File

---

I **file** sono un insieme di dati memorizzati su memoria di massa. Sono gli “archivi elettronici”.

Se il file è una successione di record con la stessa struttura si dice **file di record**, se è una successione di righe di caratteri si parla di **file di testo**.

Il modo con cui un programma utilizza i dati memorizzati dipende dalla sua organizzazione logica, ovvero dal possibile **tipo di accesso** al file, che può essere:

- **Sequenziale**
- **Diretto o random**
- **Per chiave**

# Organizzazione logica MM

## File: operazioni

---

- **Apertura:**
  - si crea un collegamento tra MM e MC associando al **nome logico** del file (riconosciuto all'interno del programma) il **nome fisico** (riconosciuto a livello di SO)
  - Si riserva in MC un buffer per le operazioni di I/O
  - Si aggiornano le tabelle di gestione dei file per specificare le informazioni necessarie per l'individuazione dei dati su MM (tabella dei file aperti)
- **lettura:** si trasferiscono i dati dalla MM al buffer relativo al file
- **scrittura:** si trasferiscono i dati dal buffer relativo al file alla MM
- **chiusura:** si aggiornano le tabelle di gestione file

# Organizzazione logica MM

## File ASCII o binari

---

Il SO tratta i file in 2 modi, come:

- **File di testo o ASCII:** ovvero come successione di caratteri ASCII, distinguendo una riga da un'altra grazie ad un carattere di fine riga.
- **File binari:** successione di byte senza distinguere una riga da un'altra. I dati possono essere organizzati nel caso di file di record.

Per poter trattare i dati memorizzati su un file, questi si devono trasferire dalla memoria di massa alla memoria centrale (operazione di *input*) o viceversa (operazione di *output*). Il trasferimento fisico è relativo non ad un singolo byte, ma ad un insieme di byte: **blocco fisico** (che, in generale, contiene 1 o più record - righe). Dal punto di vista del programmatore il trasferimento è relativo ad un solo record - riga.



# Organizzazione logica MM

## File: accesso sequenziale

---

- I file sono formati da una successione di righe o di record. I dati sono reperiti solo nell'ordine con cui sono stati memorizzati.
- È l'unico accesso possibile per i file di testo
- Esiste una funzione per riconoscere la fine del file `EndOfFile`.

Es.

File di testo, stampante trattata come un file sequenziale.

File di record ad accesso sequenziale

# Organizzazione logica MM

## File: accesso diretto

---

- I file sono formati dalla successione di record tutti della stessa lunghezza LR.
- I dati possono essere reperiti in qualunque ordine (**random**) specificando il numero di record NR. Pertanto per reperire il record desiderato si usa la formula

$$(NR-1) \times LR + 1$$

|   |    |    |    |
|---|----|----|----|
| 1 | 1  | 2  | 3  |
| 2 | 4  | 5  | 6  |
| 3 | 7  | 8  | 9  |
| 4 | 10 | 11 | 12 |

es.

$$(2-1) \times 3 + 1 = 4$$

# Organizzazione logica MM

## File: accesso per chiave

---

I file sono formati dalla successione di record tutti con la stessa struttura. È possibile distinguere un record dall'altro tramite il valore assunto da un campo (**chiave primaria**) I dati possono essere reperiti in qualunque ordine specificando il valore della chiave. Si parla di file **indicizzati**

**Chiave primaria:** insieme minimo di uno o più campi che permette di distinguere un record da un'altro in modo univoco. Ce ne deve essere una sola.

# Organizzazione logica MM

## File indicizzati

---

È possibile anche definire **chiavi secondarie**: ovvero altri campi su cui basare il reperimento del record. Possono essere o meno univoche.

L'insieme ordinato dei valori delle chiavi costituisce l'*indice* per ritrovare le informazioni nell'archivio. Pertanto, per l'individuazione del record tramite il valore delle varie chiavi, si aggiungono, al file contenente i record, tanti file **indice** quante sono le chiavi. I file indice hanno una struttura del tipo "Chiave|NumeroRecord" e sono organizzati in modo da facilitare l'individuazione dei record cercati.

Un indice che fa riferimento alla chiave primaria è un indice **primario**, sarà **secondario** se si riferisce ad una chiave secondaria.

# Organizzazione logica MM

## File indicizzati

---

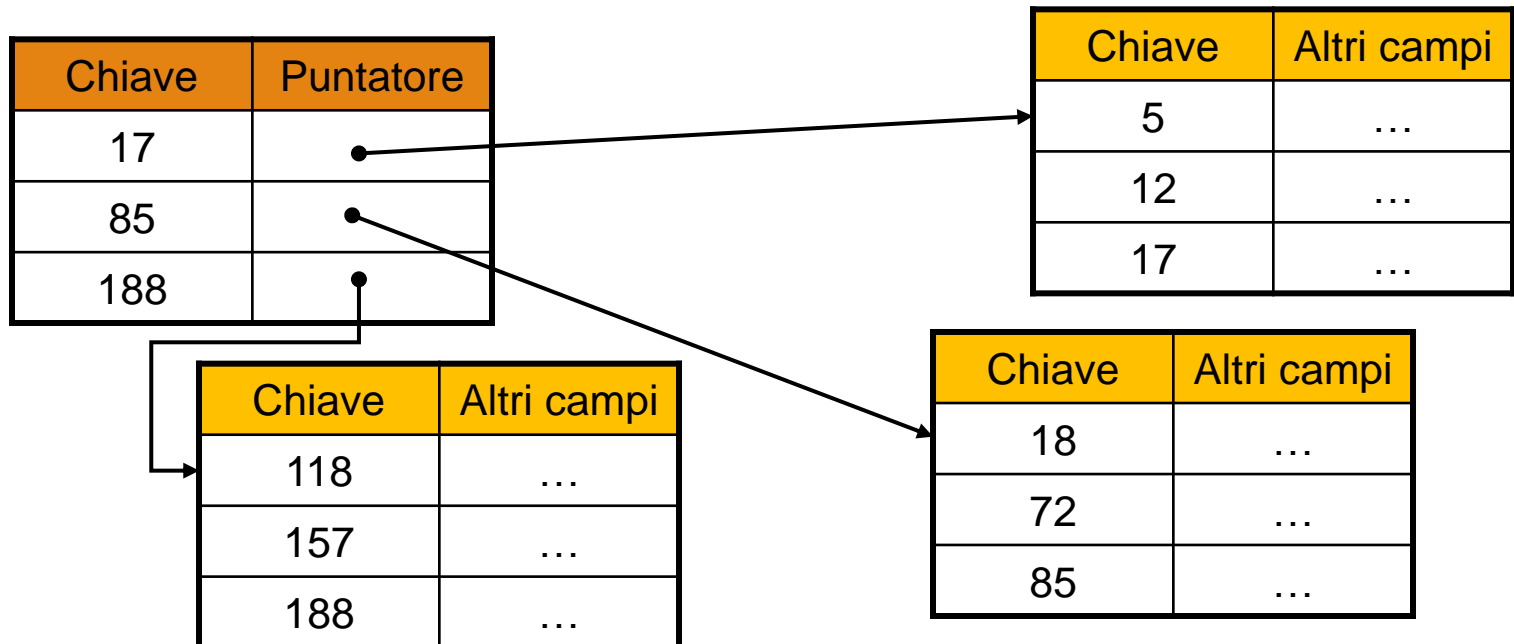
I file indicizzati sono generalmente suddivisi in 3 aree:

1. **Area Primaria:** file contenente i record, l'unica visibile da parte dell'utente
2. **Area degli Indici:** sono organizzati in modo da facilitare l'individuazione dei record cercati tramite la chiave. Sono gestiti dal SO.
3. **Area di Overflow:** spazio in MM (file o spazio dell'area primaria) utilizzato per il caricamento di dati non "inseribili" nell'area primaria.

# Organizzazione logica MM

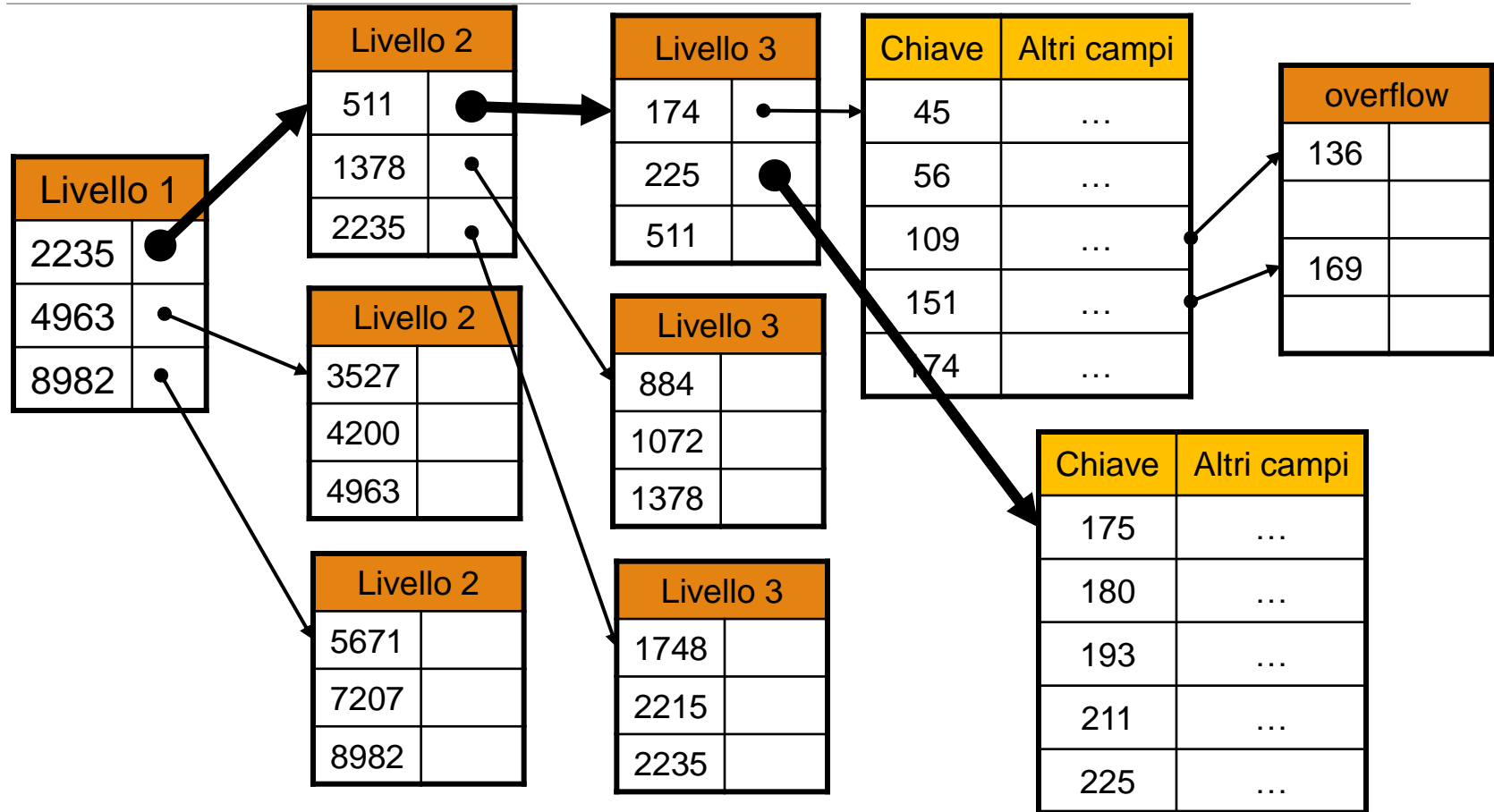
## File indicizzati: indici sparsi

L'area degli indici è generalmente organizzata ad **indici di blocco** ad 1 o più livelli. Se il file è ordinato rispetto alla chiave, si possono utilizzare indici sparsi, contenenti una tabella in cui in ogni riga c'è “*ValoreMaxBlocco | PuntatoreBlocco*”.



# Organizzazione logica MM

## File indicizzati: indici sparsi



è evidenziato il percorso per l'individuazione della chiave 180 in un **organizzazione a più livelli**

# Organizzazione logica MM

## File indicizzati: indici sparsi

---

Un organizzazione a 3 livelli è adatta a HD multipack

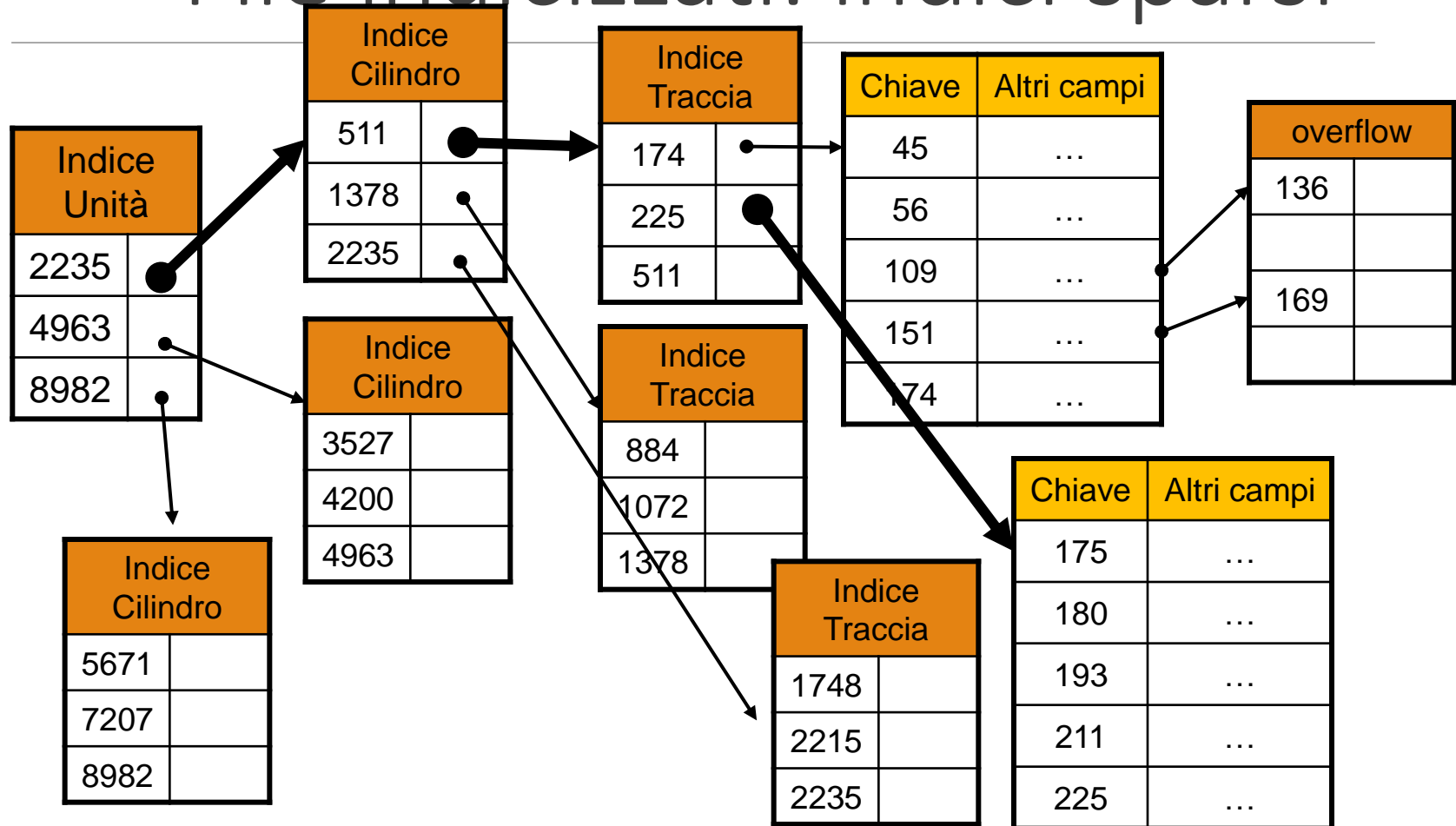
1. Se il file può essere memorizzato su più unità c'è un **indice di unità** o di disco che la specifica (allocato in modo permanente in MC)
2. In ogni unità c'è un **indice di cilindro** (nella prima traccia del primo cilindro dell'unità)
3. Per ogni cilindro c'è un **indice di traccia** (nella prima traccia del cilindro a cui si riferisce oppure tutti insieme nel 1° cilindro)

I blocchi in cui è suddiviso il file coincidono con una traccia



# Organizzazione logica MM

## File indicizzati: indici sparsi



I puntatori fanno riferimento rispettivamente al numero di: unità, cilindro, traccia, settore

# Organizzazione logica MM

## File indicizzati: overflow

---

Quando si deve inserire un record in blocco "pieno", il record viene inserito nelle aree di overflow. Queste possono essere:

- **Indipendenti:** aree alla fine di ogni cilindro o in cilindri dedicati, in cui vengono inseriti i record, ordinati logicamente tramite puntatore
- **Distribuite:** aree all'interno e alla fine di ogni blocco dell'area primaria (possono essere o meno ordinate)

Quando ricerco un record, scorro i file ad indice e i blocchi dell'area primaria in modo sequenziale cercando l'indice subito maggiore, se non lo trovo dove dovrebbe essere, vado a cercarlo nell'area di overflow del blocco stesso o in quella comune, a seconda della gestione.

Il SO può periodicamente risistemare il file in modo da svuotare l'area di overflow.

# Organizzazione logica MM

## File indicizzati: indici sparsi

---

### Considerazioni:

Il numero di voci dell'indice è uguale al *numero di blocchi su disco nel file di dati ordinato*

I valori dell'indice sono ordinati in modo che si possa fare una ricerca binaria

Dovendo mantenere i file ordinati il problema è nell'inserimento e nell'eliminazione dei record in quanto bisogna spostare i record nel file di dati, ma anche cambiare le voci dell'indice: ad ogni modifica o aggiunta, il SO deve riorganizzare le catene di puntatori.

Si può usare un file di overflow non ordinato

# Organizzazione logica MM

## File indicizzati: indici sparsi

---

Considerazioni:

Un file indice occupa meno blocchi rispetto al file dati:

1. vi sono meno voci dell'indice che record nel file dati
2. ogni voce è di dimensione inferiore rispetto a un record (ha solo due campi)

Quindi un blocco su disco può contenere più voci dell'indice che record del file dati

Quindi una ricerca sull'indice richiede meno accessi al blocco rispetto a una ricerca binaria su file dati

È usato per chiavi univoche

# Organizzazione logica MM

## File indicizzati: indici densi

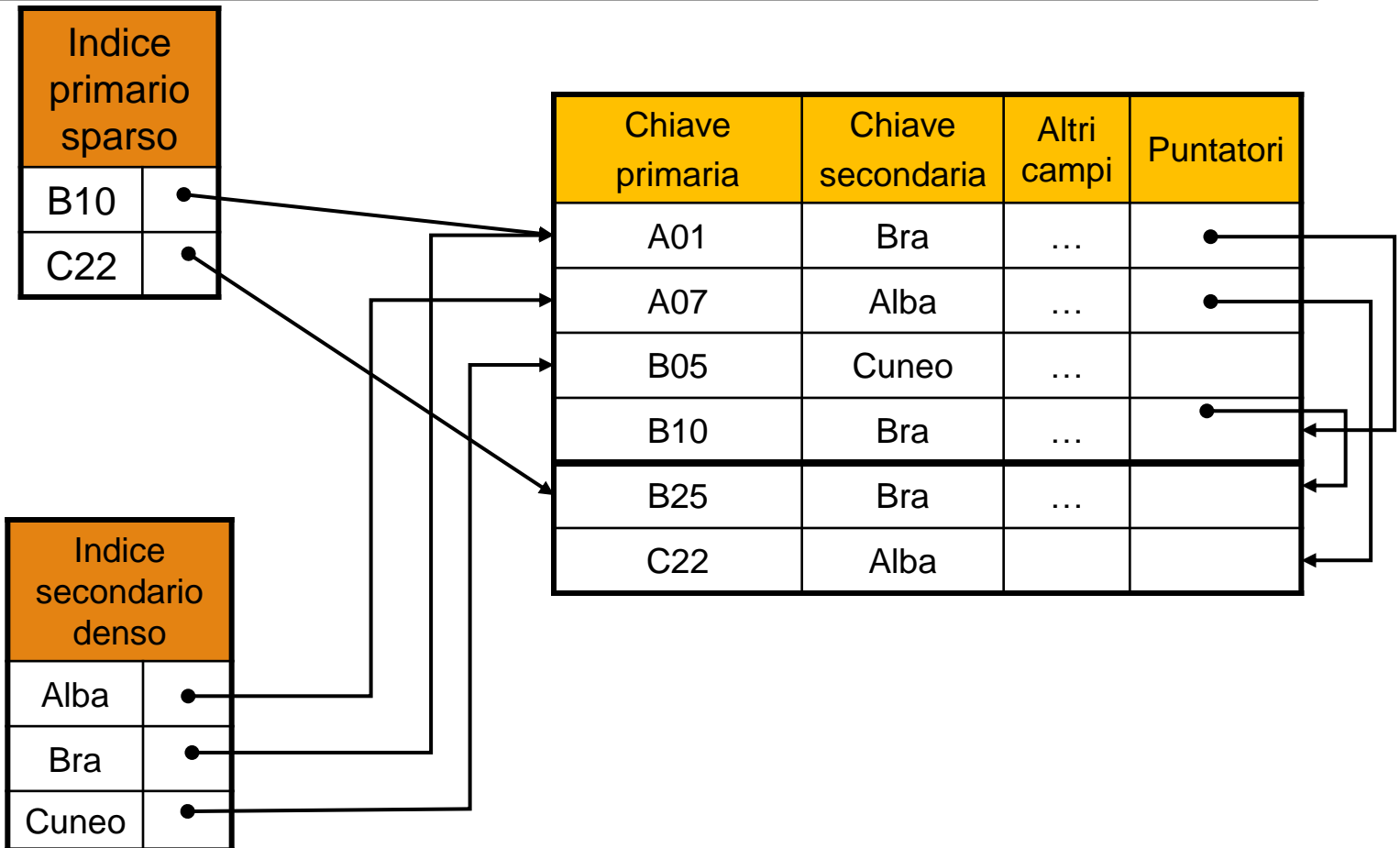
---

Quando si deve fare una ricerca per chiavi secondarie con duplicati, o quando il file non è ordinato rispetto alla chiave di ricerca, si devono utilizzare **indici densi**, che contengono tutti i possibili valori del campo a cui si riferiscono, ordinati in modo crescente. A ciascuno di questi si associa un puntatore al primo elemento avente quel valore, dal quale, tramite una catena di puntatori è possibile raggiungere tutti gli "omologhi".

Ad ogni modifica o aggiunta, il SO deve riorganizzare le catene di puntatori.

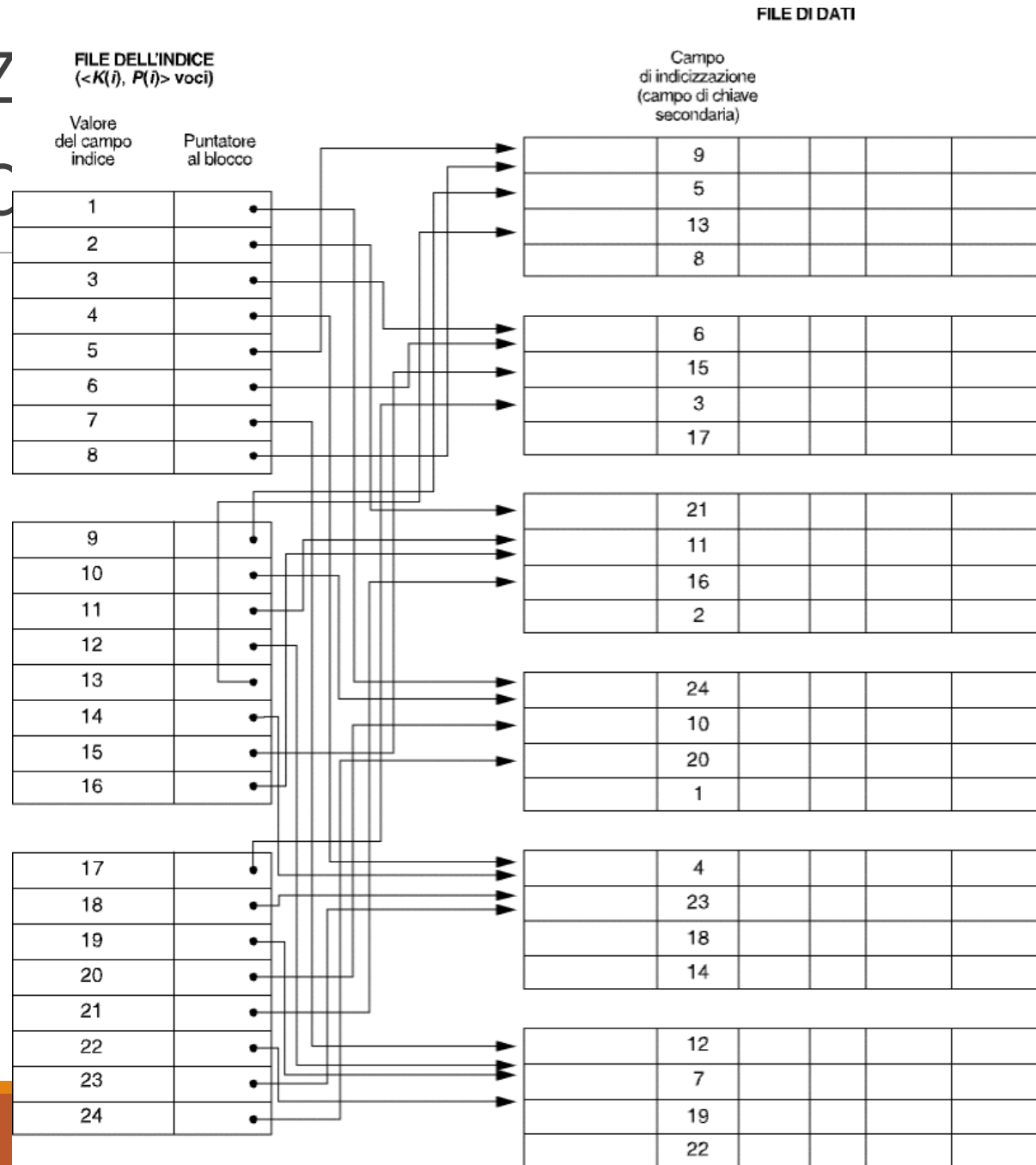
# Organizzazione logica MM

## File indicizzati: indici densi



# Organizzaz File inc

Un indice  
denso su un  
campo chiave  
sul quale non  
è stato  
effettuato un  
ordinamento  
del file



# Organizzazione logica MM

## File indicizzati: indici densi

---

### Considerazioni:

Di solito un indice denso ha bisogno di più spazio e di un tempo di ricerca maggiore rispetto a un indice sparso

Ad ogni inserimento o eliminazione si deve aggiornare il file indice e la catena di puntatori nell'area dati, ma non riordinare il file dati

Il miglioramento nel tempo di ricerca di un singolo record è comunque vantaggioso (l'alternativa sarebbe una ricerca lineare su tutto il file di dati)



# Organizzazione logica MM

## File indicizzati: B-Tree

---

Sono delle strutture di dati che permettono la rapida localizzazione dei dati dei file.

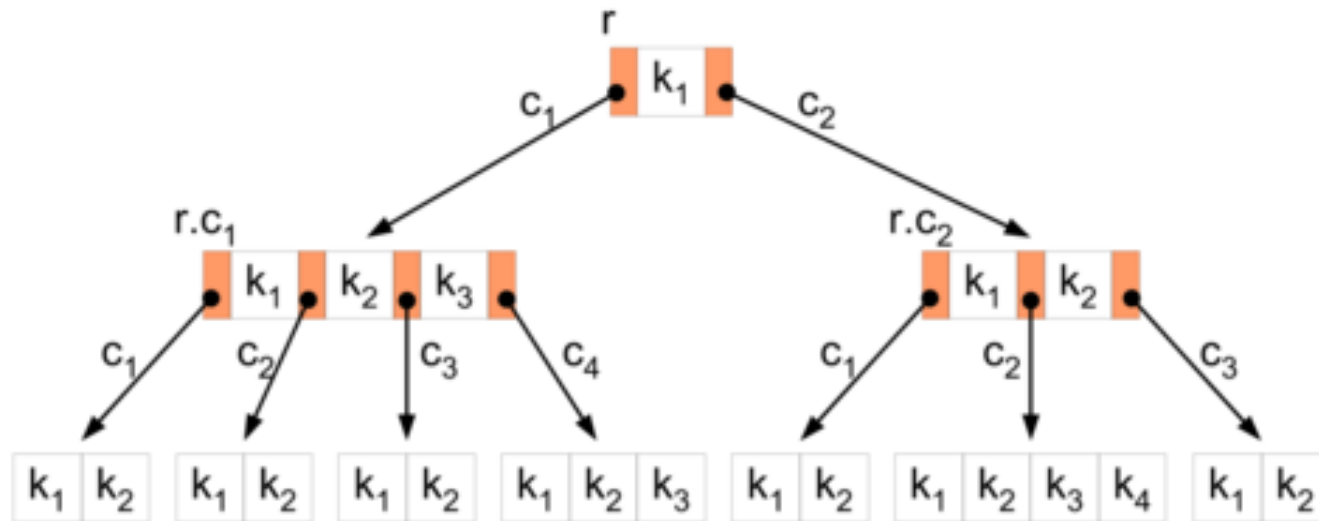
Ogni nodo contiene più chiavi e deriva dagli alberi di ricerca, in quanto ogni chiave appartenente al sottoalbero sinistro è di valore inferiore rispetto a ogni chiave appartenente ai sottoalberi alla sua destra

La loro struttura ne garantisce il bilanciamento: le altezze dei sottoalberi destro e sinistro differiscono al più di una unità.

# Organizzazione logica MM

## File indicizzati: B-Tree

Grazie alla loro organizzazione e al bilanciamento, un B-Tree permette di compiere operazioni di inserimento, cancellazione e ricerca in un file/DB in tempi ammortizzati logaritmicamente.

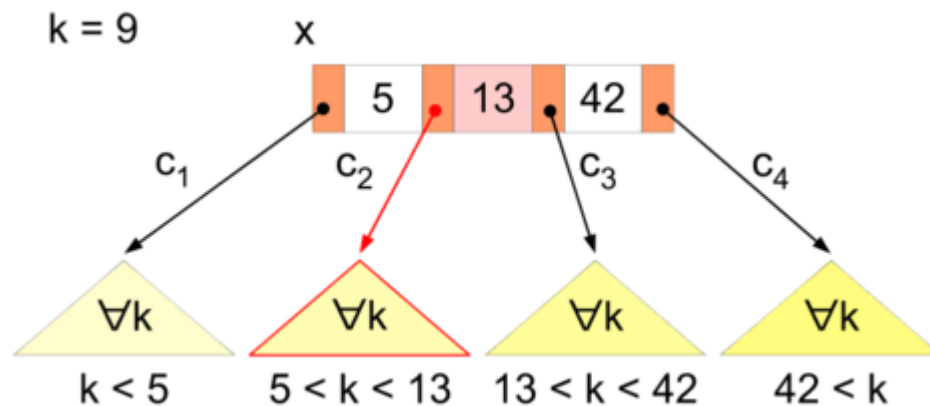


# Organizzazione logica MM

## File indicizzati: B-Tree

### Ricerca della chiave K:

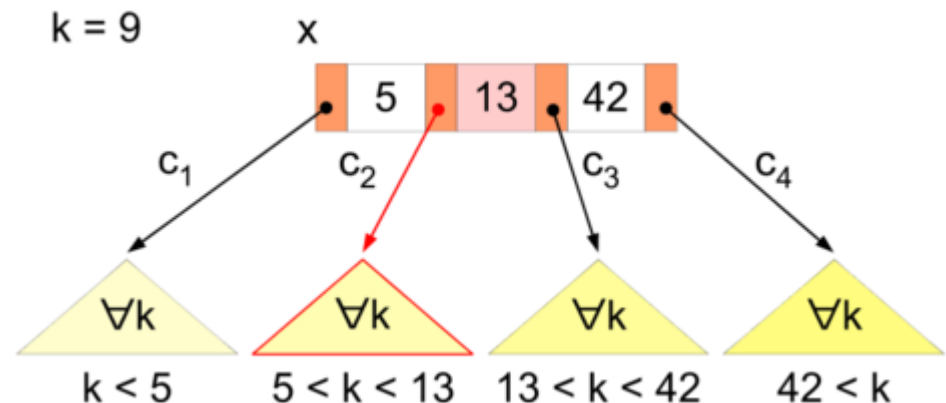
1. Trasferimento in memoria della radice
2. Ricerca di K nella pagina trasferita: se è presente, la ricerca è terminata.
3. Altrimenti, se  $K <$  chiave più a sinistra del nodo, allora trasferimento della pagina puntata dal puntatore di sinistra (se non è nullo);



# Organizzazione logica MM

## File indicizzati: B-Tree

3. se  $K >$  chiave più a destra allora trasferimento della pagina puntata dal puntatore più a destra (se non è nullo);  
se è compreso tra due chiavi del nodo allora trasferimento della pagina puntata dal puntatore compreso tra le due chiavi (se non è nullo). Ritorno al punto 2.
4. Se il puntatore in questione è nullo, la chiave non è presente.



# Organizzazione logica MM

## File indicizzati: B-Tree

---

L'inserimento o l'eliminazione di una chiave può presentare difficoltà in quanto è fondamentale mantenere l'albero bilanciato

Esistono diverse varianti al B-tree. Le tre più diffuse sono:

- Il B+tree (tutti i dati sono salvati nelle foglie. I nodi interni contengono chiavi e puntatori)
- Il B\*tree (per gestire grandi quantità di dati)
- Il prefix B-tree (evoluzione del B\*tree)
- Il predictive B+tree (è una variante del B+tree)