Strutturazione delle PAGINE che sono solitamente file sequenziali

- **DISORDINATI** → Seguono Ordine di Inserimento (INSERT EFFICIENTE)
- **ORDINATI** → Rispetto ad una CHIAVE DI ORDINAMENTO (RICERCA È OTTIMIZZATA)

Chiave Ordinamento ≠ Chiave Primaria Solitamente

per Aumentare le prestazioni Si Aggiunge la Struttura dati **INDICE**.

Per Velocizzare delle Operazioni Sui dati

Si dividono in:

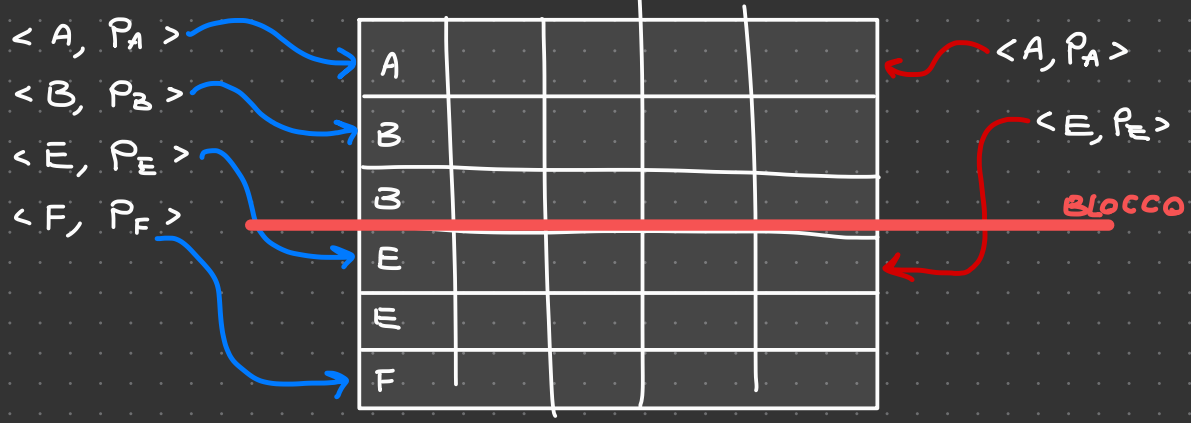
- **PRIMARIO** → file SEQ. ORDINATO **Chiave Ordinamento** = Chiave di Ricerca (Sulla Quale costruisco l'Indice)
- **SECONDARIO** → Se Sono ≠,

INDICE PRIMARIO:

Indice Come Insieme di RECORD $< V_i, p_i >$

V_i → VALORE CHIAVE DI RICERCA (= chiave Ordinamento)

p_i → PUNTATORE AL PRIMO RECORD NEL FILE CON CHIAVE V_i



Può Essere:

- **DENSO** ⇒ Si ha un RECORD per ogni possibile Valore di Chiave
  Fatto da 4 Valori di Chiave = 4 Record
- **SPARSO** ⇒ Presenti Alcuni Valoci di Chiave, il PRIMO di ogni BLOCCO

INDICE PRIMARIO SPARSO perchè è ORDINATO con la Stessa chiave

RICERCA CON INDICE PRIMARIO DENSO: (chiave K)

Scansiono Seguenzialmente i RECORD e se Trovo $<V_i=K, P_i>$ devo Seguire il puntatore per Accedere Alle TUPLE

Mi Richiede 1 Accesso All'Indice + 1 Accesso al Blocco dei dati DOVE risiedono

Se NON Lo Trovo ⇒ NON Esistono TUPLE con Quel Valore

RICERCA CON INDICE PRIMARIO SCARSO (chiave K)

Non ci Sono tutti i Valori disponibili, ma devo Cercare il RECORD $< V_i = K', p_i >$ dove K' è il più piccolo Valore di Chiave Maggiore di K. ⇒ USO RECORD PRECEDENTE

O Il duale dove K' è il più Grande Valore di K minore o uguale a K ⇒ USO DIRETTAMENTE IL PUNTATORE

Il Blocco PUNTATO da P_i potenzialmente può Contenere i dati che mi Servono Mentre con INDICE DENSO Sono sicuro che li Contiene

INSERIMENTO DI UNA TUPLA FILE SEQUENZIALE

DENSO

Chiave K è Già presente Nell' Indice? Se SI, allora NON Viene Modificato

ES: Se Aggiungo un Altro Valore di Tupla con filiale = A, Indice NON Cambia ma se fosse Stato filiale = C devo Aggiornare anche l'Indice essendo Nel Caso denso

Se Val di chiave K non è Presente Nell' Indice AGGIUNGO il Record Corrispondente

SPARSO

Se INSERIM. della Tupla causa l'Aggiunta di un NUOVO BLOCCO in memoria Secondaria allora devo Aggiungere il NUOVO RECORD all' Indice.

"1 CHIAVE V. BLOCCO"

Nell' Indice, quindi $< V_K, P_K >$

CANCELLAZIONE DI UNA TUPLA

DENSO

Cancellare il RECORD Nell' Indice Ma Solo Se la TUPLA era l'UNICA con Chiave K

SPARSO

Aggiorno l'Indice sse la Cancellazione implica la RIMOZIONE del BLOCCO.

Ma Anche SE K è presente Nell' Indice e NON ci Sono Altre Tuple con Chiave K, devo allora Aggiornare $< K, P_K >$ con $< K', P'_K >$, quindi con il Successivo Valore chiave Nel BLOCCO

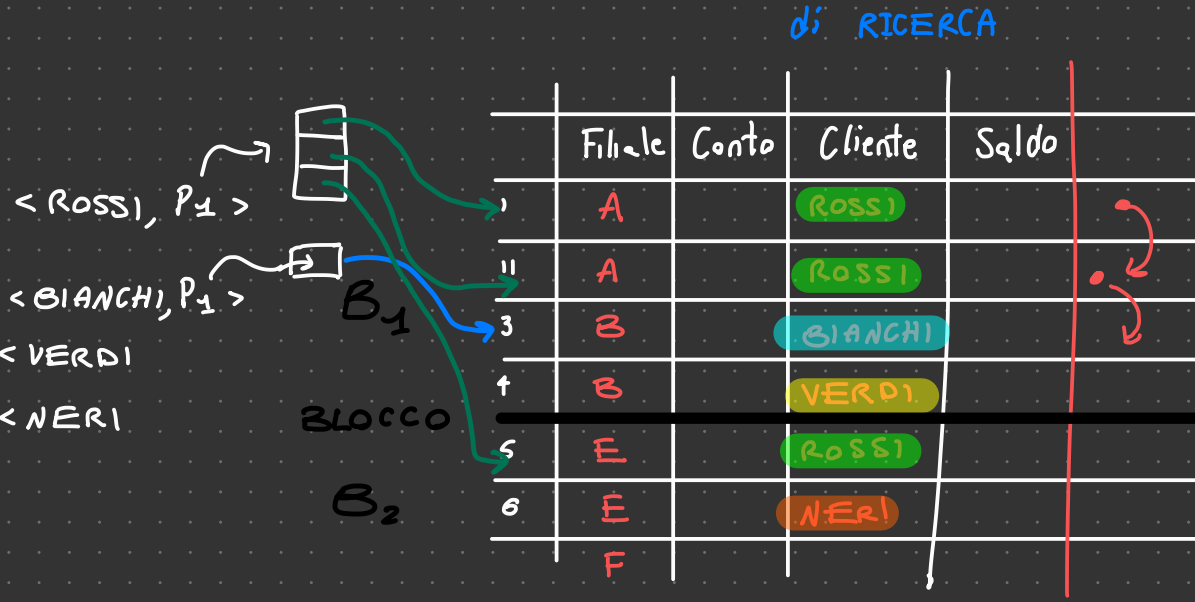
INDICI SECONDARI: (Può ESSERE Solo DENSO)

Una chiave di RICERCA che NON Coincide Con la CHIAVE di ORDINAMENTO

INDICE = Insieme di Record del TIPO $< V_i, P_i >$

Valore CHIAVE DI RICERCA

PUNTATORE MA AD UN BUCKET DI PUNTATORI che individ. Nel file SEQ. tutte le tuple Caratterizzate da



RICERCA NELL' INDICE SECONDARIO:

Scansione SEQUENZIALE dell' Indice Alla Ricerca del RECORD $<K, P_K>$.

Essendo DENSO, Se NON Trovo il RECORD allora NON C'è un TUPLA con chiave = K

Se lo Trovo, Seguo il puntatore per Accedere al BUCKET di puntatori CORRISPONDENTE.

faccio Accesso alle TUPLE Tramite i puntatori in B

Mi Richiede: 1 Accesso All' Indice + 1 Accesso al Bucket +

dipende dalla dimensione dell' INDICE

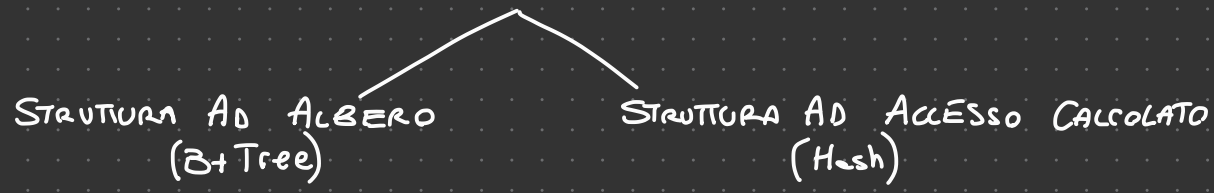
N-Accessi Alle pagine che Contengono i dati

INSERIMENTO/CANCELLAZIONE

Equivalenti per INDICE PRIMARIO DENSO Se NON per l'aggiornamento del BUCKET

Indice Solitamente << dati Tanto che Si Carica in MEMORIA CENTRALE tutto.

Ma Se i DATI Crescono potrebbe essere che l'indice NON risieda più in mem. Centrale Interamente



Indice è fatto per Velocizzare le Interrogazioni

INDICE B+ TREE:

È una Struttura ad Albero, in Cui Ogni Nodo è Memorizzato in una pagina della Memoria Secondaria

I Legami Stabiliti Tra i Nodi Sono puntatori a Pagine di Memoria Secondaria

Ogni NODO ha Molti figli ⇒ Pochi livelli e Molti NODI FOGLIA riducendo così il #pagine da CARICARE IN MEM. PRINCIPALE

È bilanciato, Lunghezza dei percorsi RADICE→FOGLIA è Uguale per Tutte le foglie. Il che Necessita OPERAZIONI Aggiuntive per Mantenerlo BILANCIATO

FAN-OUT ⇒ #Max di puntatori che può contenere un Nodo foglia

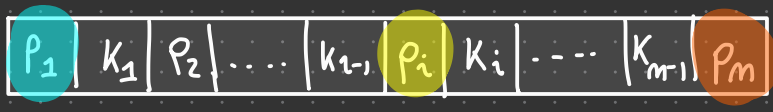
NODO FOGLIA:

fino a n-1 Valori di Chiave ed n puntatori con m≤n

PUNTA ALLA PRIMA TUPLA CON CHIAVE K_1 ⇒ Indice Primario
AL BUCKET DELLE TUPLE ⇒ Indice Secondario

Hanno un VINCOLO DI ORDINAMENTO:

L_i $\forall K_t \in L_i$ vale che $K_t < K_{t+1}$
L_i, L_{i+1} $\forall K_t \in L_i, \forall K_h \in L_{i+1}$. $K_t < K_h$

NODO INTERMEDIO: 

SOTTOALBERO CHE CONTIENE VALORI DI CHIAVE $K < K_1$

SOTTOALBERO CON Valori di Chiave K tali che $K_{i-1} \leq K < K_i$

SOTTOALBERO CON CHIAVE $K_{m-1} \leq K$

VINCOLI DI RIEMPIMENTO

Dato un FAN-OUT n:

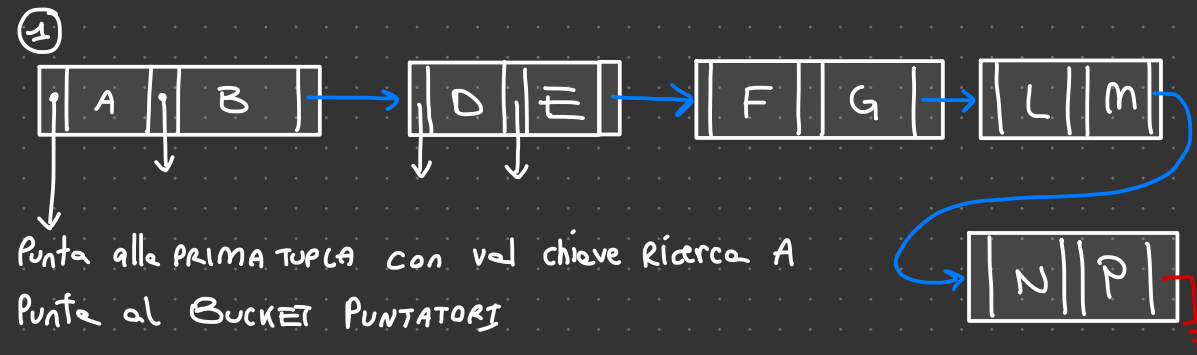
NODI FOGLIA $\Rightarrow$ $\lceil \frac{n-1}{2} \rceil \leq \#Chiavi \leq (n-1)$

NODI INTERMEDI $\Rightarrow$ $\lceil n/2 \rceil \leq \#Puntatori \leq n$

ESEMPIO: con FAN-OUT = 4

1 NODI FOGLIA $\Rightarrow$ $2 \leq \#Chiavi \leq 3$

2 NODI INTERMEDI $\Rightarrow$ $2 \leq \#puntatori \leq 4$



(2)

TUTTO ASSEME

UNICO che può VIOLARE i vincoli di RIEMPIMENTO MINIMO

K<D

D≤K≤F

K≥F

Caso con riempimento massimo: