

Strutture fisiche e strutture di accesso ai dati



ALBERTO BELUSSI

SECONDA PARTE
ANNO ACCADEMICO 2018-2019

Gestore dei metodi di accesso

2

E' il modulo del DBMS che esegue il piano di esecuzione prodotto dall'ottimizzatore e produce sequenze di accessi alle pagine della base di dati presenti in memoria secondaria.

piano di esecuzione



Gestore dei metodi d'accesso



Richieste di accesso alle pagine
DATI/INDICI (fix, unfix, setDirty)

Gestore dei metodi di accesso

3

Metodi d'accesso

Sono i moduli software che implementano gli algoritmi di accesso e manipolazione dei dati organizzati in specifiche strutture fisiche.

Esempio:

- Scansione sequenziale
- Accesso via indice
- Ordinamento
- Varie implementazioni del join

Metodi di accesso

4

Ogni metodo d'accesso ai dati conosce:

- L'organizzazione delle **tuple** nelle pagine DATI salvate in memoria secondaria (come una **tabella** viene organizzata in pagine DATI della memoria secondaria)
- L'organizzazione fisica della pagine DATI contenenti tuple e delle pagine che memorizzano le strutture fisiche di accesso o INDICI (come i record dell'indice vengono memorizzati all'interno delle pagine).

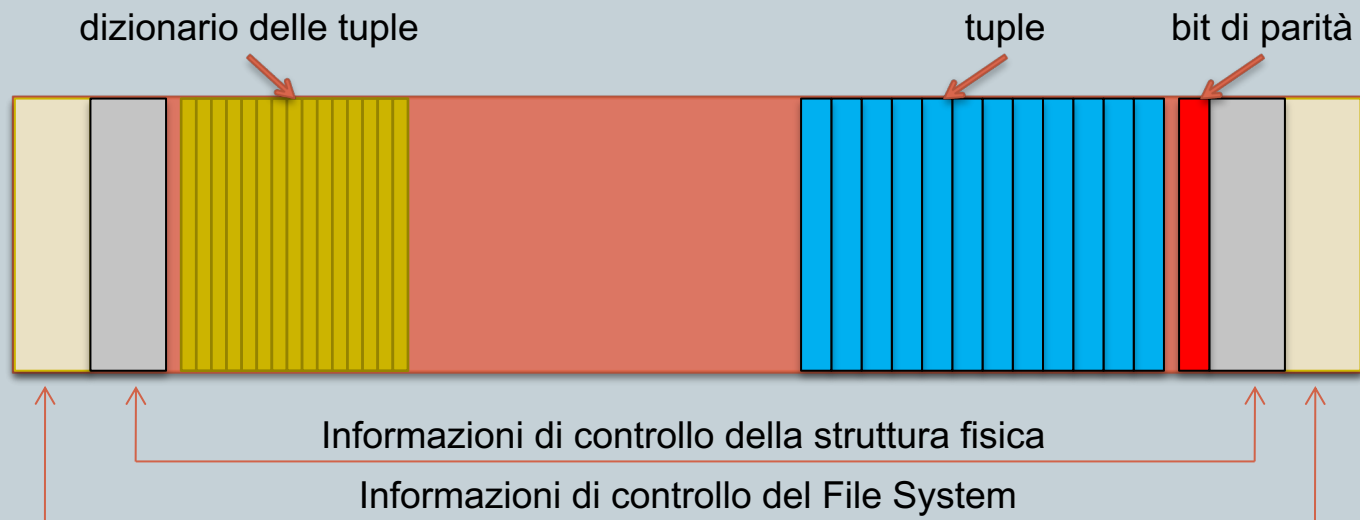
Organizzazione di una pagina DATI

5

In una pagina DATI sono presenti informazioni utili e informazioni di controllo:

- Informazioni utili: tuple della tabella
- Informazioni di controllo: dizionario, bit di parità, altre informazioni del file system o della specifica struttura fisica.

STRUTTURA della pagina DATI



Struttura del dizionario

- Tuple di lunghezza fissa: il dizionario non è necessario, si deve solo memorizzare la dimensione delle tuple e l'offset del punto iniziale
- Tuple di lunghezza variabile: il dizionario memorizza l'offset di ogni tupla presente nella pagina e di ogni attributo di ogni tupla

Lunghezza massima di una tupla = dimensione massima dell'area disponibile su una pagina, altrimenti va gestito il caso di tuple memorizzate su più pagine (in postgresQL si veda la soluzione TOAST - The Oversized-Attribute Storage Technique) .

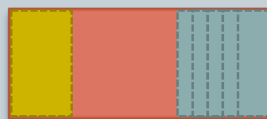
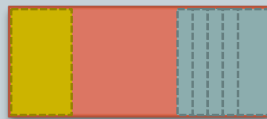
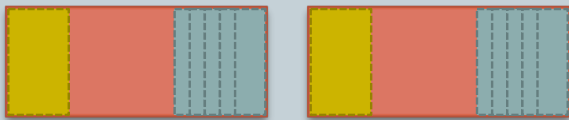
Operazioni

- Inserimento di una tupla
 - Se esiste spazio contiguo sufficiente: inserimento semplice
 - Se non esiste spazio contiguo ma esiste spazio sufficiente: riorganizzare lo spazio ed eseguire un inserimento semplice
 - Se non esiste spazio sufficiente: operazione rifiutata
- Cancellazione: sempre possibile anche senza riorganizzare
- Accesso ad una tupla
- Accesso ad un attributo di una tupla
- Accesso sequenziale (di solito in ordine di chiave primaria)
- Riorganizzazione

Rappresentazione di una tabella a livello fisico

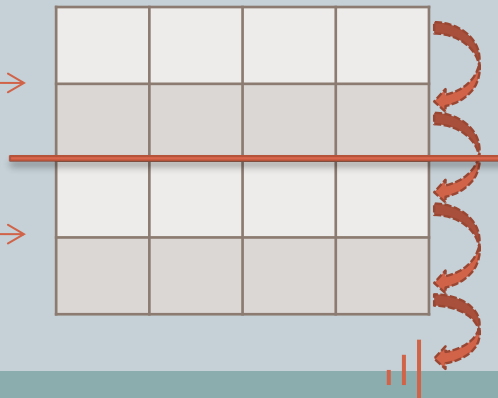
8

Livello fisico (memoria secondaria)



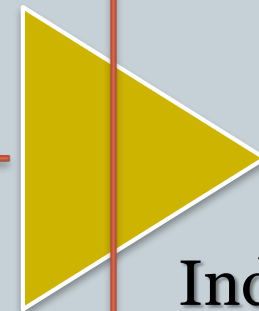
Struttura FISICA:

- Strutture ad accesso calc.
- Seriale
- Array
- SEQUENZIALE ORDINATA
o FILE SEQUENZIALE



Livello logico (modello dei dati relazionale)

Tabella



Indice

Strutture fisiche di rappresentazione dei dati

9

File sequenziale (struttura sequenziale ordinata in base alla chiave di ordinamento)

Caratteristica fondamentale: è un file sequenziale dove le tuple sono ordinate secondo una chiave di ordinamento.

Esempio

Pagina 1

Filiale	Conto	Cliente	Saldo	
A	102	Rossi	1000	
B	110	Rossi	3020	
B	198	Bianchi	500	

Pagina 2

E	17	Neri	345	
E	102	Verdi	1200	
E	113	Bianchi	200	
H	53	Neri	120	
F	78	Verdi	3400	



FILE SEQUENZIALE

10

Operazioni

- **Inserimento di una tupla**
 - Individuare la pagina P che contiene la tupla che precede, nell'ordine della chiave, la tupla da inserire.
 - Inserire la tupla nuova in P; se l'operazione non va a buon fine aggiungere una nuova pagina (overflow page) alla struttura: la pagina contiene la nuova tupla, altrimenti si prosegue.
 - Aggiustare la catena di puntatori.
- **Scansione sequenziale ordinata secondo la chiave (seguendo i puntatori)**
- **Cancellazione di una tupla**
 - Individuare la pagina P che contiene la tupla da cancellare.
 - Cancellare la tupla da P.
 - Aggiustare la catena di puntatori.
- **Riorganizzazione:** si assegnano le tuple alle pagine in base ad opportuni coefficienti di riempimento, riaggiustando i puntatori.

INDICI

11

Per aumentare le prestazioni degli accessi alle tuple memorizzate nelle strutture fisiche (FILE SEQUENZIALE), si introducono strutture ausiliarie (dette strutture di accesso ai dati o INDICI).

Tali strutture velocizzano l'accesso casuale via chiave di ricerca. La chiave di ricerca è un insieme di attributi utilizzati dall'indice nella ricerca.

Indici su file sequenziali

- INDICE PRIMARIO: in questo caso la **chiave di ordinamento** del file sequenziale coincide con la **chiave di ricerca** dell'indice.
- INDICE SECONDARIO: in questo caso invece la **chiave di ordinamento** e la **chiave di ricerca** sono diverse.

INDICE PRIMARIO

12

Usa una chiave di ricerca che coincide con la chiave di ordinamento del file sequenziale.

Ogni record dell'indice primario contiene una coppia $\langle v_i, p_i \rangle$:

- v_i : valore della chiave di ricerca;
- p_i : puntatore al primo record nel file sequenziale con chiave v_i

Esistono due varianti dell'indice primario:

- INDICE DENSO: per ogni occorrenza della chiave presente nel file esiste un corrispondente record nell'indice.
- INDICE SPARSO: solo per alcune occorrenze della chiave presenti nel file esiste un corrispondente record nell'indice, tipicamente una per pagina.

INDICE PRIMARIO

13

Esempio

Indice denso

A	
B	
E	
H	
M	

Pagina 1

A	
E	

Pagina 2

Indice sparso

Filiale	Conto	Cliente	Saldo	
A	102	Rossi	1000	
B	110	Rossi	3020	
B	198	Bianchi	500	
E	17	Neri	345	
E	102	Verdi	1200	
E	113	Bianchi	200	
H	53	Neri	120	
M	78	Verdi	3400	

INDICE PRIMARIO

14

Operazioni

- Ricerca di una tupla con chiave di ricerca K .
 - DENSO ($\Rightarrow K$ è presente nell'indice)
 - ✦ Scansione sequenziale dell'indice per trovare il record (K, p_k)
 - ✦ Accesso al file attraverso il puntatore p_kCosto: 1 accesso indice + 1 accesso pagina dati
 - SPARSO ($\Rightarrow K$ potrebbe non essere presente nell'indice)
 - ✦ Scansione sequenziale dell'indice fino al record $(K', p_{k'})$ dove K' è il valore più grande che sia minore o uguale a K
 - ✦ Accesso al file attraverso il puntatore $p_{k'}$ e scansione del file (pagina corrente) per trovare le tuple con chiave K .Costo: 1 accesso indice + 1 accesso pagina dati

INDICE PRIMARIO

15

Operazioni

- Inserimento di un record nell'indice

Come inserimento nel FILE SEQUENZIALE (nella pagina della memoria secondaria invece di tuple ci sono record dell'indice)

- DENSO

- ✦ L'inserimento nell'indice avviene solo se la tupla inserita nel file ha un valore di chiave K che non è già presente.

- SPARSO

- ✦ L'inserimento avviene solo quando, per effetto dell'inserimento di una nuova tupla, si aggiunge una pagina dati alla struttura; in tutti gli altri casi l'indice rimane invariato.

INDICE PRIMARIO

16

Operazioni

- Cancellazione di un record nell'indice

Come cancellazione nel FILE SEQUENZIALE

- DENSO

- ✦ La cancellazione nell'indice avviene solo se la tupla cancellata nel file è l'ultima tupla con valore di chiave K.

- SPARSO

- ✦ La cancellazione nell'indice avviene solo quando K è presente nell'indice e la corrispondente pagina viene eliminata; altrimenti, se la pagina sopravvive, va sostituito K nel record dell'indice con il primo valore K' presente nella pagina.

INDICE SECONDARIO

17

Usa una chiave di ricerca che **NON** coincide con la chiave di ordinamento del file sequenziale.

Ogni record dell'indice secondario contiene una coppia $\langle v_i, p_i \rangle$:

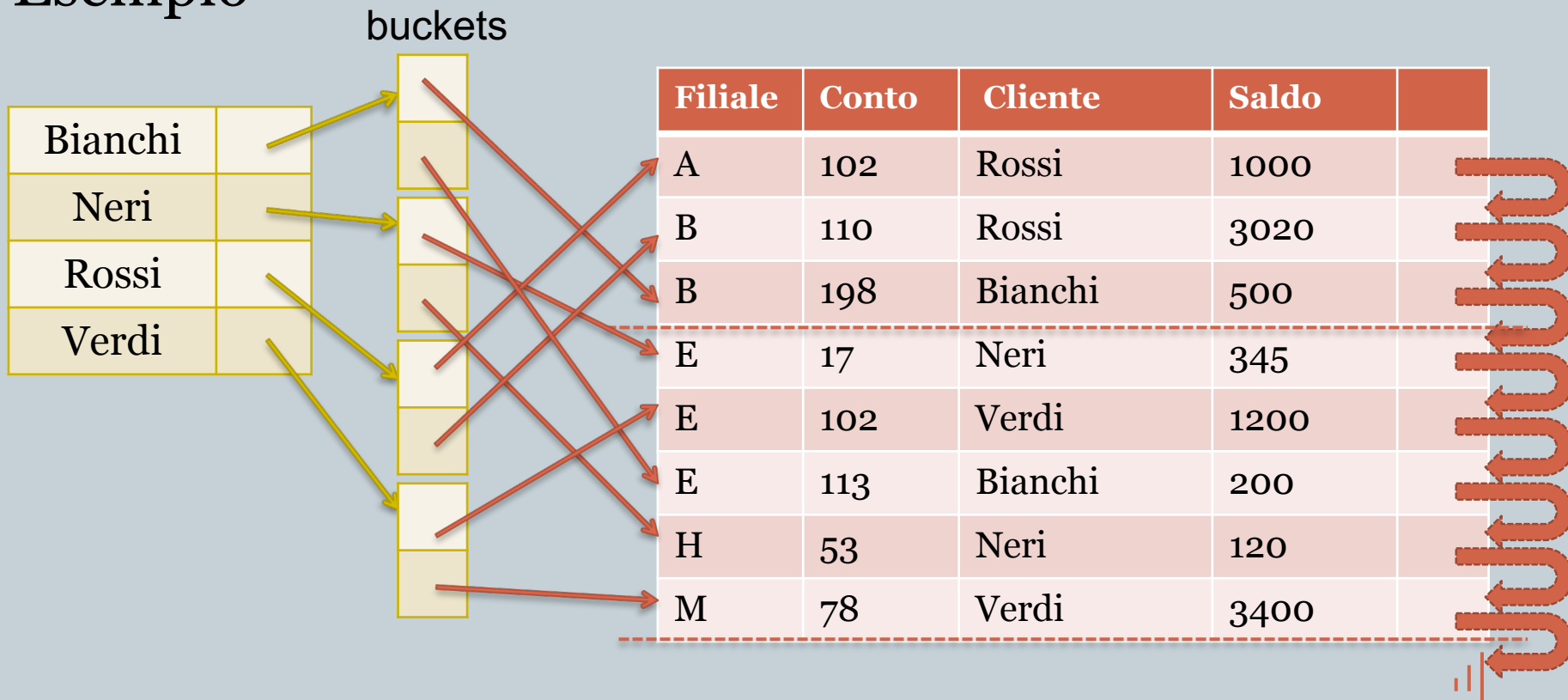
- v_i : valore della chiave di ricerca;
- p_i : puntatore al **bucket di puntatori** che individuano nel file sequenziale tutte le tuple con valore di chiave v_i .

Gli indici secondari sono sempre **DENSI**.

INDICE SECONDARIO

18

Esempio



INDICE SECONDARIO

19

Operazioni

- Ricerca di una tupla con chiave di ricerca K .
 - ✦ Scansione sequenziale dell'indice per trovare il record (K, p_k)
 - ✦ Accesso al bucket B di puntatori attraverso il puntatore p_k
 - ✦ Accesso al file attraverso i puntatori del bucket B .

Costo: 1 accesso indice + 1 accesso al bucket + n accessi pagine dati
- Inserimento e cancellazione: come indice primario denso con in più l'aggiornamento dei bucket.