

# Gestione delle Strutture Ausiliarie di Accesso (Indici)

Lorenzo Vaccarecci

15 Marzo 2024

## 1 Esempio

File heap

colloc	dataNol	codCli	dataRest
1111	01-Mar-2006	6635	02-Mar-2006
1115	01-Mar-2006	6635	02-Mar-2006
1118	10-Mar-2006	6642	11-Mar-2006
1117	02-Mar-2006	6635	06-Mar-2006
1118	02-Mar-2006	6635	06-Mar-2006
1111	04-Mar-2006	6642	05-Mar-2006
1119	08-Mar-2006	6635	10-Mar-2006
1120	08-Mar-2006	6635	10-Mar-2006
1116	08-Mar-2006	6642	09-Mar-2006
1121	15-Mar-2006	6635	18-Mar-2006
1122	15-Mar-2006	6635	18-Mar-2006
1113	15-Mar-2006	6635	18-Mar-2006
1129	15-Mar-2006	6635	20-Mar-2006
1119	15-Mar-2006	6642	16-Mar-2006
1126	15-Mar-2006	6610	16-Mar-2006
1112	16-Mar-2006	6610	18-Mar-2006
1114	16-Mar-2006	6610	17-Mar-2006
1128	18-Mar-2006	6642	20-Mar-2006
1124	20-Mar-2006	6610	21-Mar-2006
1115	20-Mar-2006	6610	21-Mar-2006
1124	21-Mar-2006	6642	22-Mar-2006
1117	21-Mar-2006	6610	?
1127	22-Mar-2006	6635	?
1125	22-Mar-2006	6635	?
1122	22-Mar-2006	6642	?
1113	22-Mar-2006	6642	?
1116	22-Mar-2006	6610	?

File ordinato su dataNol

colloc	dataNol	codCli	dataRest
1111	01-Mar-2006	6635	02-Mar-2006
1115	01-Mar-2006	6635	02-Mar-2006
1117	02-Mar-2006	6635	06-Mar-2006
1118	02-Mar-2006	6635	06-Mar-2006
1111	04-Mar-2006	6642	05-Mar-2006
1119	08-Mar-2006	6635	10-Mar-2006
1120	08-Mar-2006	6635	10-Mar-2006
1116	08-Mar-2006	6642	09-Mar-2006
1118	10-Mar-2006	6642	11-Mar-2006
1121	15-Mar-2006	6635	18-Mar-2006
1122	15-Mar-2006	6635	18-Mar-2006
1113	15-Mar-2006	6635	18-Mar-2006
1129	15-Mar-2006	6635	20-Mar-2006
1119	15-Mar-2006	6642	16-Mar-2006
1126	15-Mar-2006	6610	16-Mar-2006
1112	16-Mar-2006	6610	18-Mar-2006
1114	16-Mar-2006	6610	17-Mar-2006
1128	18-Mar-2006	6642	20-Mar-2006
1124	20-Mar-2006	6610	21-Mar-2006
1115	20-Mar-2006	6610	21-Mar-2006
1124	21-Mar-2006	6642	22-Mar-2006
1116	21-Mar-2006	6610	?
1117	21-Mar-2006	6610	?
1127	22-Mar-2006	6635	?
1125	22-Mar-2006	6635	?
1122	22-Mar-2006	6642	?
1113	22-Mar-2006	6642	?

File hash su dataNol

colloc	dataNol	codCli	dataRest
1117	02-Mar-2006	6635	06-Mar-2006
1118	02-Mar-2006	6635	06-Mar-2006
1111	04-Mar-2006	6642	05-Mar-2006
1119	08-Mar-2006	6635	10-Mar-2006
1120	08-Mar-2006	6635	10-Mar-2006
1116	08-Mar-2006	6642	09-Mar-2006
1111	01-Mar-2006	6635	02-Mar-2006
1115	01-Mar-2006	6635	02-Mar-2006
1118	10-Mar-2006	6642	11-Mar-2006
1121	15-Mar-2006	6635	18-Mar-2006
1122	15-Mar-2006	6635	18-Mar-2006
1113	15-Mar-2006	6635	18-Mar-2006
1129	15-Mar-2006	6635	20-Mar-2006
1119	15-Mar-2006	6642	16-Mar-2006
1126	15-Mar-2006	6610	16-Mar-2006
1112	16-Mar-2006	6610	18-Mar-2006
1114	16-Mar-2006	6610	17-Mar-2006
1124	20-Mar-2006	6610	21-Mar-2006
1115	20-Mar-2006	6610	21-Mar-2006
1128	18-Mar-2006	6642	20-Mar-2006
1124	21-Mar-2006	6642	22-Mar-2006
1116	21-Mar-2006	6610	?
1117	21-Mar-2006	6610	?
1127	22-Mar-2006	6635	?
1125	22-Mar-2006	6635	?
1122	22-Mar-2006	6642	?
1113	22-Mar-2006	6642	?

Supponiamo di voler recuperare tutti i noleggi effettuati in una certa data nel file ordinato su *dataNol*.

- A **livello logico**: descrizione richiesta in modo dichiarativo
- A **livello fisico**: algoritmo che implementa a livello fisico la richiesta, ad esempio:
  - apri il file che contiene i record
  - doppio while per il file e per il blocco di dati nel file dove controlliamo se la data è quella che ci interessa

Ho dovuto leggere tutti i blocchi di dati prima della data per trovare i record che mi interessano. *Bene ma non benissimo: basti pensare a un file con migliaia di record.*

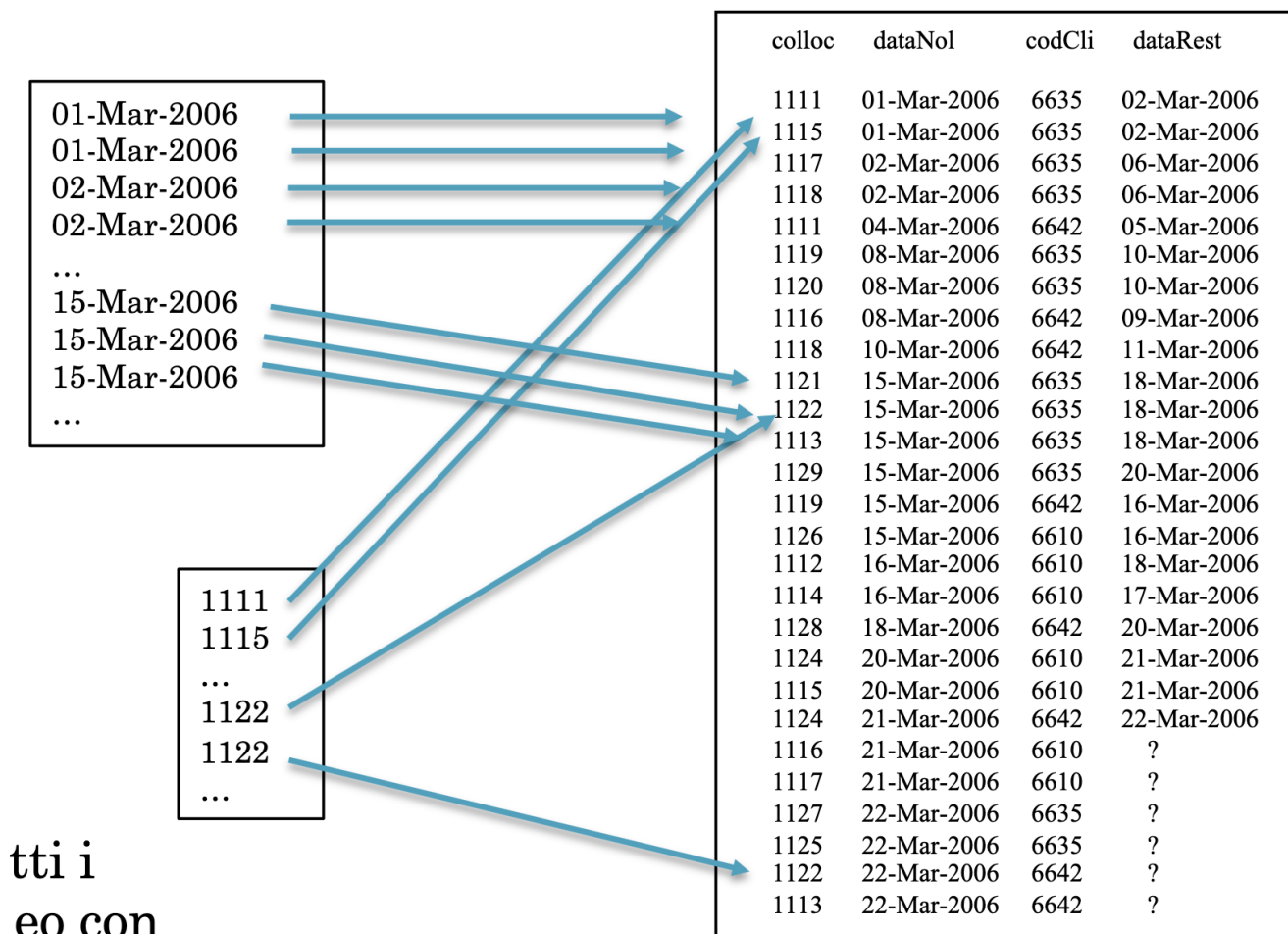
Supponiamo ora di voler recuperare tutti i noleggi effettuati con un certo colloc nel file ordinato su *dataNol*.

- A **livello logico**: descrizione richiesta in modo dichiarativo
- A **livello fisico**: algoritmo che implementa a livello fisico la richiesta, ad esempio:
  - apri il file che contiene i record
  - doppio while per il file e per il blocco di dati nel file dove controlliamo se colloc è quello che ci interessa

Ho dovuto leggere tutti i blocchi di dati per trovare i record che mi interessano. *Non efficiente, devo leggere tutto il file.*

## 2 Perchè gli indici

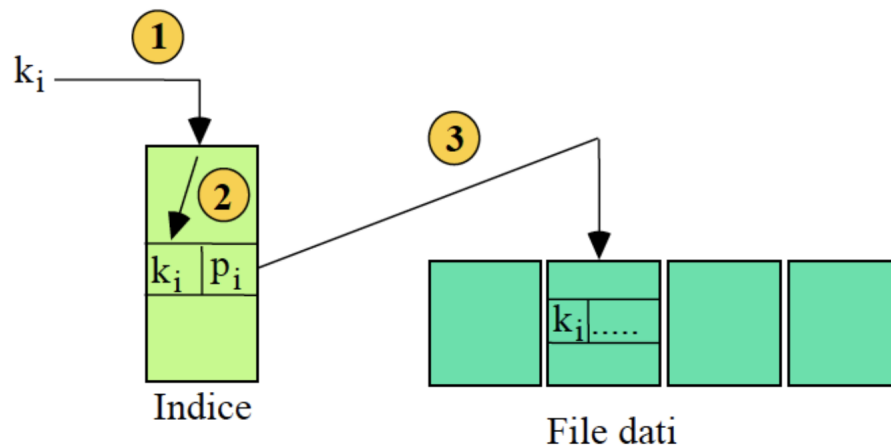
Sono delle strutture ausiliare create dal sistema per velocizzare l'accesso ai dati. Gli indici riferiti a una chiave ordinata è detto **clusterizzato**, altrimenti **non clusterizzato**.



Un indice su una relazione  $R$  è un insieme di coppie del tipo  $\{(k_i, r_i), \dots, (k_n, r_n)\}$  dove  $k$  è un valore  $x$  un attributo (o un insieme di attributi) chiamato chiave di ricerca e  $r$  è un puntatore al file dei dati.

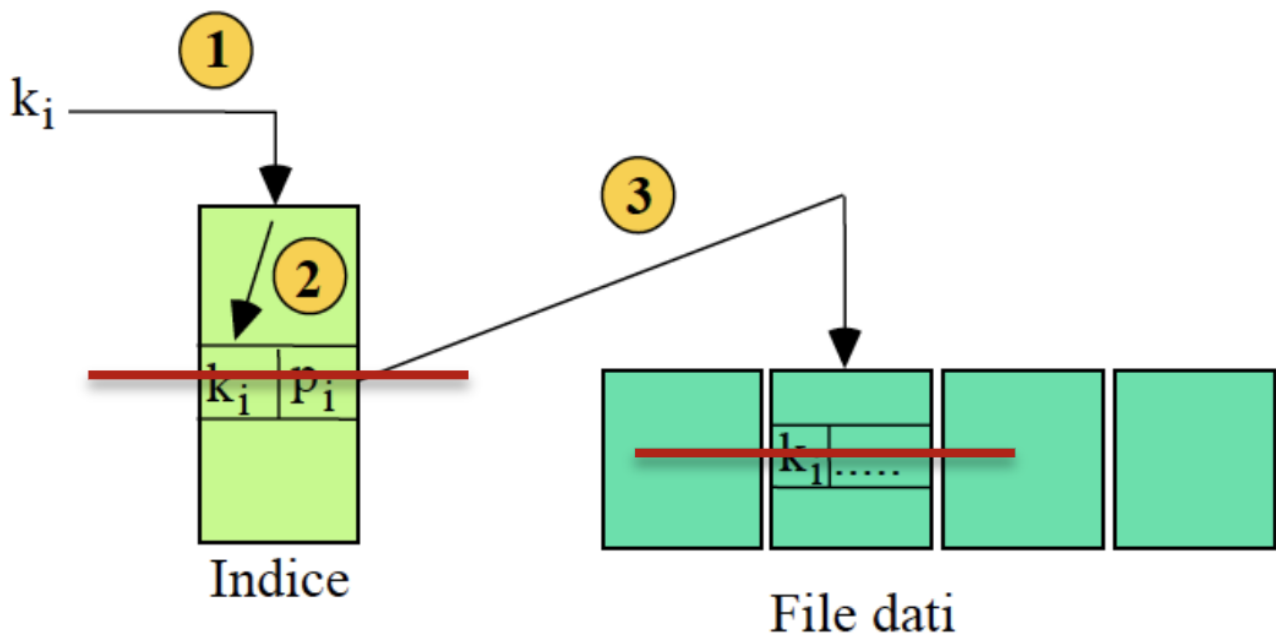
### 3 Accesso con indice

1. Accedere all'indice
2. Ricercare la coppia  $(k_i, p_i)$
3. Accedere alla pagina dati relativa



### 4 Indici e Aggiornamenti

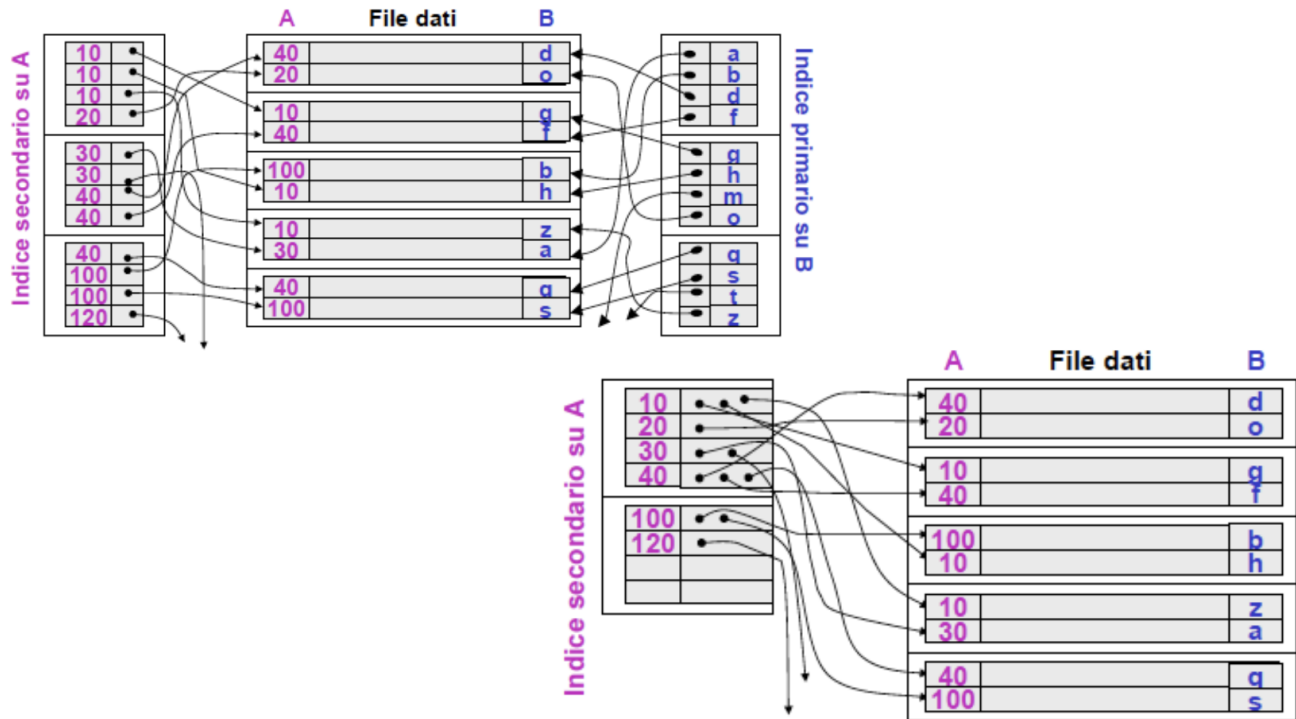
Gli indici rendono efficiente l'esecuzione delle interrogazioni ma rendono più complessi gli aggiornamenti (ogni volta che si aggiorna il file dei dati bisogna aggiornare anche l'indice).



## 5 Indici primari e secondari

**Primari** : se la chiave di ricerca soddisfa il vincolo di unicità

**Secondari** : se la chiave di ricerca non soddisfa il vincolo di unicità, per evitare di replicare inutilmente lo stesso valore si raggruppano le chiavi associando una lista di puntatori.



## 6 Tipi di indici

**Ordinati** : o indici ad albero, le coppie  $(k_i, r_i)$  vengono salvate in un file ordinate rispetto ai valori della chiave  $k_i$

**Hash** : le coppie non vengono memorizzate su un file ma "calcolate" attraverso l'uso di una funzione hash  $h(k_i) = r_i$