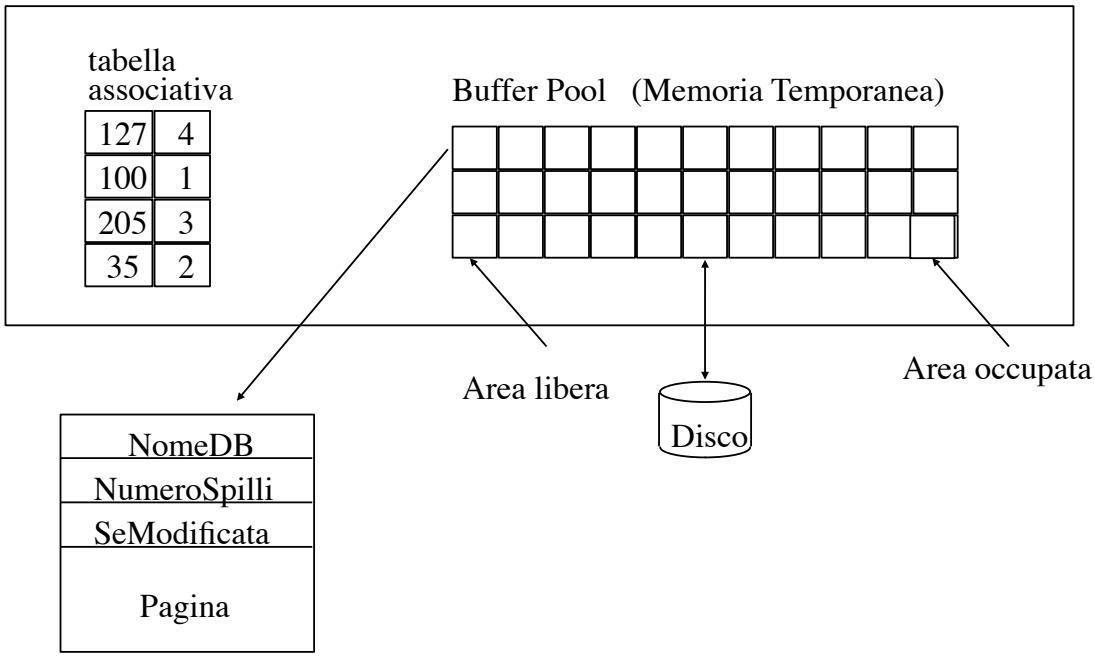
GESTORE MEMORIA PERMANENTE E GESTORE DEL BUFFER

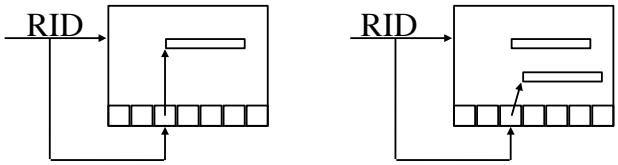
- GESTORE MEMORIA PERMANENTE
  - Fornisce un'astrazione della memoria permanente in termini di insiemi di file logici di pagine fisiche di registrazioni (blocchi), nascondendo le caratteristiche dei dischi e del sistema operativo.
- GESTORE DEL BUFFER
  - Si preoccupa del trasferimento delle pagine tra la memoria temporanea e la memoria permanente, offrendo agli altri livelli una visione della memoria permanente come un insieme di pagine utilizzabili in memoria temporanea, astraendo da quando esse vengano trasferite dalla memoria permanente al buffer e viceversa

GESTORE DEL BUFFER: AREA DELLE PAGINE



STRUTTURA DI UNA PAGINA

- Struttura fisica: un insieme, di dimensione fissa, di caratteri .
- Struttura logica:
  - informazioni di servizio;
  - un'area che contiene le stringhe che rappresentano i record;
- Il problema dei riferimenti ai record: coppia (PID della pagina, posizione nella pagina) (RID).



GESTORE STRUTTURE DI MEMORIZZAZIONE

- Tipi di organizzazioni:
  - Seriali o Sequenziali
  - Per chiave
  - Per attributi non chiave
- Parametri che caratterizzano un'organizzazione:
  - Occupazione di memoria
  - Costo delle operazioni di:
    Ricerca per valore o intervallo
    Modifica
    Inserzione
    Cancellazione

ORGANIZZAZIONI SERIALE E SEQUENZIALE

- Organizzazione seriale (heap file): i dati sono memorizzati in modo disordinato uno dopo l'altro:

  Semplice e a basso costo di memoria

  Poco efficiente

  Va bene per pochi dati

  - E' l'organizzazione standard di ogni DBMS.

- Organizzazione sequenziale: i dati sono ordinati sul valore di uno o più attributi:

  Ricerche più veloci

  Nuove inserzioni fanno perdere l'ordinamento

ORDINAMENTO DI ARCHIVI

- Perché è importante l'ordinamento di archivi
  - Risultato di interrogazioni ordinato (order by)
  - Per eseguire alcune operazioni relazionali (join, select distinct, group by)
- Algoritmo merge-sort: costo N*Log(N)

ORGANIZZAZIONE PER CHIAVE

- **Obiettivo**: mantiene i dati ordinati, permette di trovare il record con un valore di una chiave, oppure tutti i record con i valore della chiave compreso con un certo intervallo
- **Alternative**:
  - Organizzazione procedurale (hash) o ad albero (indice)
  - Organizzazione statica o dinamica.

ORGANIZZAZIONE PROCEDURALE (HASH)

- **Algoritmo di trasformazione della chiave** : data una chiave, restituisce l'indirizzo della pagina in cui memorizzare, e successivamente cercare, il record
- **Proprietà**:
  - Metodo di trasformazione dall'insieme delle chiavi a quello degli indirizzi di pagina
  - Parametri di dimensionamento
  - Gestione delle collisioni e dei trabocchi
  - Organizzazione statica o dinamica.

ORGANIZZAZIONE AD ALBERO

- **Struttura ad albero**: tutte le registrazioni sono memorizzate (ordinate) in una struttura ad albero generalizzazione dell'albero di ricerca bilanciato (B^+-albero)
- **Proprietà**:
  - Permette di trovare il record con un valore della chiave A con pochi accessi
  - Permette di trovare record che soddisfano condizioni del tipo $A \leq v$, $A \geq v$, $v1 \leq A \leq v2$
  - Mantiene i dati ordinati

INDICI

- Si usa un indice, ovvero di un insieme **ordinato** di coppie (k, r(k)), dove k è un valore della chiave ed r(k) è un riferimento al record con chiave k.
- L'indice è gestito di solito con un **B^+-albero**, la struttura più usata e ottimizzata dai DBMS.
- Gli indici possono essere multi-attributi.
- Gli indici possono essere su attributi non chiave.

ESEMPI DI INDICI PER ATTRIBUTO CHIAVE O NON CHIAVE

- Tabella:

RID	Matr	Prov	An
1	106	MI	1972
2	102	PI	1970
3	107	PI	1971
4	104	FI	1968
5	100	MI	1970
6	103	PI	1972

- Indici

Matr	RID
100	5
102	2
103	6
104	4
106	1
107	3

Indice su Matr

An	RID
1968	4
1970	2
1970	5
1971	3
1972	1
1972	6

Indice su An

SCELTA DELL'ORGANIZZAZIONE

- È una parte fondamentale della progettazione fisica dei dati, può portare a prestazioni radicalmente diverse per l'esecuzione delle query
- Compito difficile per la necessità di ottimizzare le prestazioni globali
- Si usano strumenti di supporto, può cambiare durante la vita del db
- È necessario fare una scelta appropriata degli indici
- È necessario conoscere le caratteristiche del DBMS che si usa

ORGANIZZAZIONI NEI DBMS

- Tutti i DBMS memorizzano una relazione con l'organizzazione seriale se non specificato altrimenti

- Si possono aggiungere (e togliere) indici con il comando:
  CREATE [UNIQUE] INDEX Nome ON Rel(Att...)

- Si possono specificare indici al momento della creazione di tabella
  Oracle:
  CREATE TABLE R(Pk Tipo PRIMARY KEY, . . .) ORGANIZED INDEX;

- SQL Server:
  CREATE TABLE R(Pk Tipo PRIMARY KEY, . . .)
  CREATE CLUSTERED INDEX RAlbero ON R(Pk)

REALIZZAZIONE DEGLI OPERATORI RELAZIONALI

- Si considerano i seguenti operatori:
  - Proiezione
  - Selezione
  - Raggruppamento
  - Join

PROIEZIONE


```
SELECT   DISTINCT
         Provincia
FROM     Studenti R
```

- Approccio basato sull'ordinamento (non è l'unico!):
  - Si legge R e si scrive T che contiene solo gli attributi della SELECT
  - Si ordina T su tutti gli attributi
  - Si eliminano i duplicati

RESTRIZIONE CON CONDIZIONE SEMPLICE


```
SELECT  *
FROM    Studenti R
WHERE   R.Provincia = 'PI'
```

- Senza indice e dati disordinati: Npag(R).
- Con indice (B^+-albero): CI + CD

OPERAZIONI DI AGGREGAZIONE

- Senza GROUP BY
  - Si visitano i dati e si calcolano le funzioni di aggregazione.
- Con GROUP BY
  - Approccio basato sull'ordinamento (non è l'unico!):

    Si ordinano i dati sugli attributi del GROUP BY, poi si visitano i dati e si calcolano le funzioni di aggregazione per ogni gruppo.

GIUNZIONE


```
SELECT  *
FROM    Studenti S, Esami E
WHERE   S.Matricola=E.Matricola
```

- R x S è grande; pertanto, R x S seguito da una restrizione è inefficiente.