Organizzazione fisica dei dati

Index

Memorie a stato solido

Nelle memorie a stato solido, rispetto ai dischi rigidi, l'organizzazione dei dati è molto simile (blocchi, settori, …) ma non ci sono organi in movimento e i tempi di accesso e archiviazione sono ridotti.
Si lavora infatti nell'ordine dei decimi di millisecondo, mentre il tempo di accesso dei dischi magnetici è oltre 50 volte maggiore. In ogni caso, il tempo di trasferimento dati, soprattutto in scrittura è più lento del tempo di elaborazione della CPU

Blocchi e accessi a memoria secondaria

Al momento della formattazione del disco, ogni traccia è suddivisa in **blocchi** (o pagine) di **dimensione fissa** (compresa tra 2^9 e 2^{12} byte).
Per **accesso** si intende il trasferimento di un blocco da memoria secondaria a memoria

principale (**lettura** di un blocco) o da memoria principale a memoria secondaria (**scrittura** di un blocco)

Accessi a memoria secondaria

Il tempo per un accesso è dato dalla somma di:

- tempo di posizionamento → della testina sulla traccia in cui si trova un blocco
- ritardo di rotazione → rotazione necessaria per posizionare la testina all'inizio del blocco
- tempo di trasferimento → dei dati contenuti nel blocco

Il tempo richiesto per un accesso a memoria secondaria è nell'ordine di millisecondi, quindi notevolmente superiore a quello di elaborazione dei dati in memoria principale. Per questo motivo il **costo di un'operazione** sulla base di dati è definito in termini di **numero di accessi**

Memorizzazione di relazioni

Una relazione viene così trattata quando viene memorizzata:

- Ad ogni **relazione** corrisponde un *file di record* che hanno tutti lo stesso tipo (numero e tipo di campi)
- Ad ogni **attributo** corrisponde un *campo*
- Ad ogni **tupla** corrisponde un *record*
  Nel modello relazionale in ogni file ci sono record appartenenti ad un'unica relazione

---

Record

In un record oltre ai campi che corrispondono agli attributi, ci possono essere altri campi che contengono **informazioni sul record** stesso o **puntatori** ad altri record/blocchi

Informazioni sul record

All'inizio di un record alcuni byte possono essere utilizzati per:

- specificare il tipo di record → è necessario quando in uno stesso blocco sono memorizzati record di tipo diverso, cioè appartenenti a più file
- specificare la lunghezza del record → se il record ha campi a lunghezza variabile
- contenere un bit di "cancellazione" o un bit di "usato/non usato"

Per accedere ad un campo

Se tutti i campi del record hanno **lunghezza fissa**, l'inizio di ciascun campo sarà sempre ad un numero fisso di byte dall'inizio del record. Questo numero viene indicato con l'*offset* (numero di byte che precedono il campo)

Se il record contiene campi a **lunghezza variabile** si hanno due opzioni:

- all'inizio di ogni campo c'è un contatore che specifica la lunghezza del campo in numero di byte
- all'inizio del record ci sono gli offset dei campi a lunghezza variabile (tutti i campi a lunghezza fissa precedono quelli a lunghezza variabile)

> **Info**
>
> Nel primo caso per individuare la posizione di un campo bisogna esaminare i campi precedenti per vedere quanto sono lunghi; quindi la prima strategia è meno efficiente della seconda

Puntatori

Un puntatore ad un record/blocco è un dato che permette di accedere rapidamente al record/blocco; lo si può fare in due maniere

- si utilizza l'indirizzo dell'inizio (primo byte) del record/blocco sul disco
- nel caso di un record, si usa una coppia (b, k) dove b è l'indirizzo del blocco che contiene il record mentre k è il valore della chiave

> **Info**
>
> Nel secondo caso è possibile spostare il record all'interno del blocco, nel primo no, altrimenti potremmo avere dei *dangling pointer*

Blocco

In un blocco ci possono essere **informazioni sul blocco** stesso e/o **puntatori** ad altri blocchi

Informazioni sul blocco

Se un blocco contiene solo record di **lunghezza fissa** allora il blocco è diviso in **aree** (sottoblocchi) di lunghezza fissa, ciascuna delle quali può contenere un record mentre i bit “usato/non usato” sono raccolti in uno o più byte all’inizio del blocco

> **Info**
>
> Se bisogna inserire un record nel blocco occorre cercare un’area non usata; se il bit “usato/non usato” è in ciascun record ciò può richiedere la scansione di tutto il blocco; per evitare ciò si possono raccogliere tutti i bit “usato/non usato” in uno o più byte all’inizio del blocco

Se un blocco contiene record di **lunghezza variabile** si hanno due opzioni:

- si pone in ogni record un campo che ne specifica la lunghezza in termini di numero di byte
- si pone all’inizio del blocco una **directory contenente i puntatori (offset) ai record del blocco**

Directory

La directory può essere realizzata in uno dei modi seguenti:

- è preceduta da un campo che specifica quanti sono i puntatori nella directory
- è una lista di puntatori (la fine della lista è specificata da uno 0)
- ha dimensione fissa e contiene il valore 0 negli spazi che non contengono puntatori

---

Operazioni sulla base di dati

Un’operazione sulla base di dati consiste di:

- **ricerca**
- **inserimento** (implica la ricerca se vogliamo evitare duplicati)

- **cancellazione** (implica ricerca)
- **modifica** (implica ricerca)
  di un record

> **Info**
>
> Notare che la ricerca è alla base di tutte le altre operazioni

Un requisito fondamentale di un DBMS è l'**efficienza**, cioè la capacità di rispondere alle richieste dell'utente il più rapidamente possibile
Una particolare **organizzazione fisica** dei dati (cioè una particolare organizzazione dei record nei file) può rendere efficiente l'elaborazione di particolari richieste, quindi l'amministratore della base di dati durante il progetto fisico della base di dati deve tener presente quali operazioni saranno effettuate più frequentemente

Generalmente ad ogni oggetto base di un modello logico (schemi di relazione nel modello relazionale, segmenti nel modello gerarchico, tipi di record nel modello a rete) corrisponde un file di record che hanno tutti lo stesso formato, cioè gli stessi campi (che corrispondono agli attributi nel caso del modello relazionale e ai campi di segmenti e di tipi di record negli altri due modelli)

> **Warning**
>
> Stesso formato non significa stessa lunghezza, ma stesso numero e tipo di campi