

In preparazione dell'esame

# Basi dati II: programma

- Algebra relazionale
- Gestione della memoria secondaria e del buffer
- Organizzazione fisica dei dati
- Gestione ("ottimizzazione") delle interrogazioni
- Controllo della affidabilità
- Controllo della concorrenza
- ~~Normalizzazione~~
- ~~Introduzione NoSQL~~

esercizi

esercizi

esercizi

esercizi

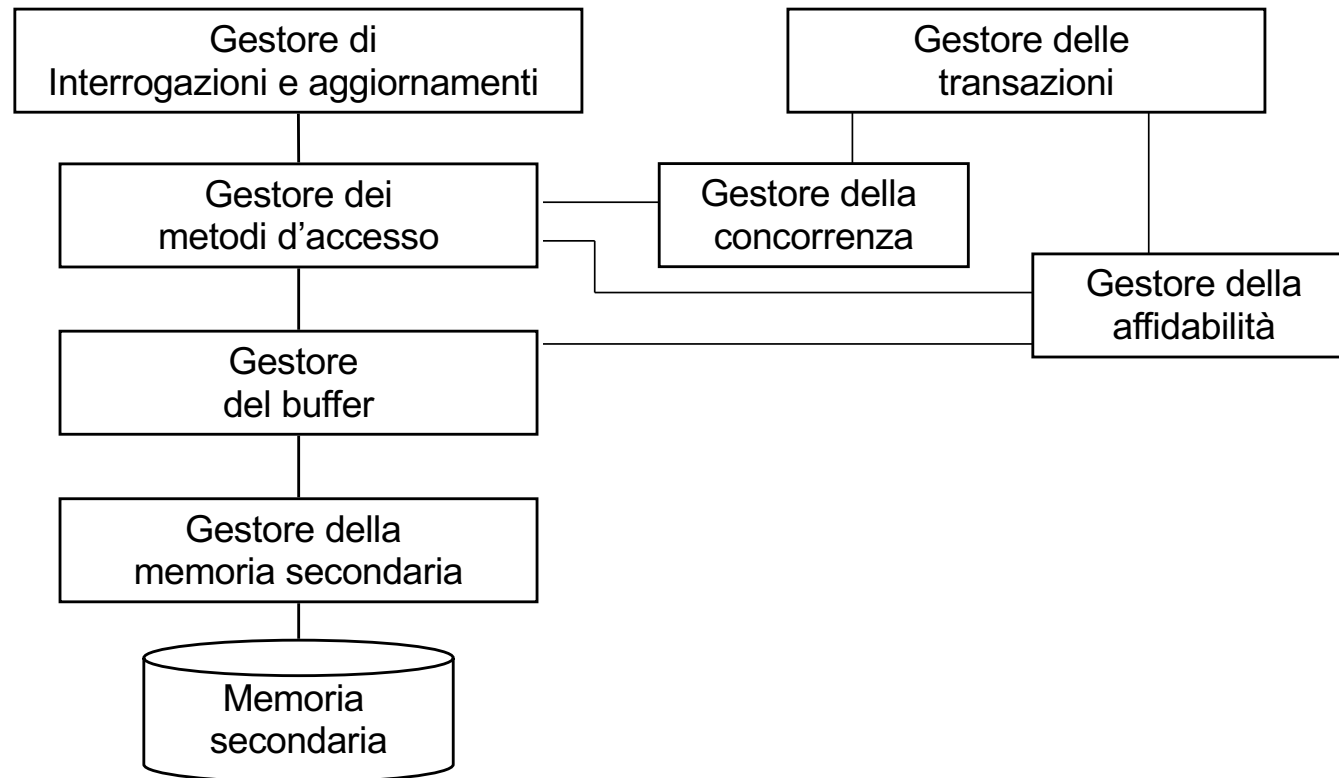
esercizi

esercizi

esercizi

## Gestore degli accessi e delle interrogazioni

## Gestore delle transazioni



# Esame

- Scritto di 2 ore, 6 domande - E' possibile utilizzare calcolatrice semplice (no cellulare)

**1 Domanda** Quiz (1pt x quiz): 4-5 Quiz (risposta multipla, risposta singola, vero/falso): esempi

- Q1 Se il grado minimo di un B-tree è 50, qual è il numero minimo di chiavi e il numero massimo di figli per ogni nodo?
  - a) 49, 100
  - b) 99, 100
  - c) 50, 50
  - d) 50, 100
- Q2 Considerate questo schema relazionale
  - dipendente(codice\_dip, nome\_dip, città\_dip)
  - reparto(codice\_rep, nome\_rep, città\_rep)
  - lavora(codice\_dipendente<sup>dipendente</sup>, codice\_reparto<sup>reparto</sup>, ore)
  - Cosa restituisce questa espressione algebrica:

$$\pi_{\text{codice\_dipendente}}(\text{Lavora} \bowtie_{\text{codice\_reparto} = \text{codice\_rep}}(\sigma_{\text{città\_rep} = \text{"Varese"}}(\text{reparto})) \cap \pi_{\text{codice\_dip}}(\sigma_{\text{città\_dip} = \text{"Varese"}}(\text{dipendente}))$$

- a) Dipendenti che vivono a Varese e lavorano in un reparto di Varese.
- b) Dipendenti che lavorano in qualche reparto di Varese, ma non vivono a Varese.
- c) Dipendenti che vivono a Varese, ma non lavorano in reparti di Varese.
- d) Dipendenti che lavorano in tutti i reparti di Varese, ma non vivono a Varese

# Esame

- Scritto di 2 ore, 6 domande - E' possibile utilizzare calcolatrice semplice (no cellulare)

4 Esercizi (5-7pt). Esercizi sui seguenti argomenti

- Organizzazione file
  - Tempi di ricerca con diverse organizzazione di file (non ordinata, ordinata, hash)
- Indici
  - Primario
  - Clustering
  - Secondario
  - Multilivello
  - B-tree
  - B<sup>+</sup>-tree
- Algebra relazionale
  - Definizione Query
  - Ottimizzazioni di espressioni/alberi di interrogazioni
- Merge sort esterno
- Costo operazioni
  - Select
  - Join
- Gestione affidabilità
  - Gestione log, ripresa a caldo e freddo
- Gestione concorrenza
  - Ricerca anomalie
  - Verifica VSR, CSR, 2PL

# Dubbi su esercizi svolti: ricerca lineare

Si considera una tabella  $R(A, C, \dots)$  con 1.000.000 di record di lunghezza fissa di 100 byte. In  $R$ ,  $A$  è campo chiave. La dimensione di  $A$  è 4 byte, la dimensione di  $C$  è 4 byte. La memoria secondaria è organizzata in blocchi di dimensione  $B = 1000$  byte.

Considerare il seguente carico di lavoro:

Op1. inserimento di un record (con verifica del vincolo di chiave), con frequenza giornaliera  $f_1 = 1$ ;

Op2. ricerca di un record sulla base del valore della chiave  $A$ , con frequenza giornaliera  $f_2 = 1000$ ;

Op3. ricerca di un record sulla base del valore dell'attributo  $C$ , con frequenza giornaliera  $f_3 = 10.000$ ;

Assumere, inoltre, che di notte il sistema esegue la riorganizzazione delle strutture fisiche

Definire i costi giornalieri valutando le seguenti soluzioni

(A) Struttura primaria ordinata sull'attributo  $A$ , con eventuali inserimenti effettuati in coda al file

(B) Struttura primaria non ordinata con indice  $B^+$ -tree sull'attributo  $A$

(C) Struttura primaria ordinata sull'attributo  $A$  con indice  $B^+$ -tree su campo  $C$ , con eventuali inserimenti effettuati in coda al file

(D) Struttura primaria non ordinata, con indice  $B^+$ -tree sull'attributo  $A$  e indice  $B^+$ -tree sull'attributo  $C$

Si assuma, indici  $B^+$ -tree sono usati con fattore di riempimento a 70%, un puntatore  $P$  al blocco occupa 6 byte, un puntatore  $P_r$  al record occupa 7 byte. Come costo stimare il tempo di esecuzione in termini di accessi alla memoria secondaria.

**Op3. ricerca di un record sulla base del valore dell'attributo C, con frequenza giornaliera  $f_3 = 10.000$**

Il file non è ordinato sull'attributo C, quindi la ricerca è sequenziale. **In media**, è necessario accedere a metà dei blocchi:

ricerca sequenziale:  $100.000/2 = 50.000$

Op3. 50.000 accessi,  $f_3=10.000 \Rightarrow 50.000*10.000 = 500.000.000$

In questo caso non abbiamo informazioni su C. Sappiamo che non è chiave, ma potrebbe essere unique. Se non specificato nel testo ipotizzate voi: in questo caso si è ipotizzato Unique.

# Dubbi su esercizi svolti: stima costo query

Si consideri la relazione  $R(\underline{A}, B, C, \dots)$  con la seguente configurazione

- $T(R) = 100.000.000$  tuple, ognuna di 40 byte
- A di 4 byte, con valori consecutivi da 1 a 100.000.000
- $B = 4000$ , il blocco B ha dimensione 4000 byte
- $P = 6$  byte, puntatore al blocco occupa 6 byte
- $P_r = 7$ , puntatore al record occupa 7 byte

Indicare la stima di costo per la seguente operazione:

1. `SELECT * FROM R WHERE A >= 1000 AND A <= 4000`

Considerando che file è ordinato su A, con indice B+-tree su A con fattore di riempimento al 70%



# Dubbi su esercizi svolti

SELECT \* FROM R WHERE A >= 1000 AND A <=4000

file è ordinato su A, con indice B+-tree su A con fattore di riempimento al 70%

La stima di costo è la somma di 3 costi:

(A) Costo B+Tree per cercare puntatore a A=1000:

47 I/O

(B) Costo lettura blocchi nodi foglia B+-tree (per A >= 1000 AND A <=4000):

4 I/O

12 I/O

(C) costo lettura delle singole tuple nei file dati:

31 I/O

Devo leggere 3001 tuple su file ordinato. Le tuple sono memorizzate contigue, quindi

Fattore blocco:  $4000/40=100$

3001 tuple contigue richiedono di leggere  $3001/100= 31$  blocchi

## Dubbi su esercizi svolti: B<sup>+</sup>-tree

- Si supponga di costruire un B<sup>+</sup>-tree su un campo di ricerca V (9 byte), usando blocchi B (512 byte), con puntatore al blocco P (6 byte) e puntatori ai record Pr (7 byte). Si ipotizzi, inoltre, che ogni nodo sia completo al 70% (fattore riempimento)
- Quanti record possono essere indicizzati da un B<sup>+</sup>-tree a 3 livelli?
  - Calcolo l'ordine del B<sup>+</sup>-tree:  $p_{\text{interno}}$ ,  $p_{\text{foglia}}$
  - Stimo numero dei valori di ricerca con fattore riempimento

# Dubbi su esercizi svolti: B<sup>+</sup>-tree

- Si supponga di costruire un B<sup>+</sup>-tree su un campo di ricerca V (9 byte), usando blocchi B (512 byte), con puntatore al blocco P (6 byte) e puntatori ai record Pr (7 byte). Si ipotizzi, inoltre, che ogni nodo sia completo al 70% (fattore riempimento)
- Calcolo l'ordine massimo per il **nodo interno** ( $p_{\text{interno}}$ )
- Un nodo interno di B<sup>+</sup>-tree può avere fino a **p** puntatori ai nodi e **p-1** valori del campo

$$p * P + (p-1) * V \leq 512$$

$$p * 6 + (p-1) * 9 \leq 512$$

$$6p + 9p - 9 \leq 512$$

$$15p \leq 521$$

Nel B<sup>+</sup>-tree il valore massimo  $p_{\text{interno}} = 34$

# Dubbi su esercizi svolti: B<sup>+</sup>-tree

- Si supponga di costruire un B<sup>+</sup>-tree su un campo di ricerca V (9 byte), usando blocchi B (512 byte), con puntatore al blocco P (6 byte) e puntatori ai record Pr (7 byte). Si ipotizzi, inoltre, che ogni nodo sia completo al 70% (fattore riempimento)
- Calcolo l'ordine massimo per il **nodo foglia** ( $p_{\text{foglia}}$ )
- Un nodo foglia di B<sup>+</sup>-tree contiene fino a **p** valori del campo e **p** puntatori ai record, più un puntatore al nodo successivo

$$p * Pr + p * V + P \leq 512$$

$$p * 7 + p * 9 + 6 \leq 512$$

$$16p \leq 506$$

Nel B<sup>+</sup>-tree il valore massimo  $p_{\text{foglia}} = 31$