732G16: Databaser; Design och programmering

Övning

Fysisk design av databasen

Transaktioner

2024

LINKÖPINGS UNIVERSITET

Innehåll

- Räkna på utrymme för en tabell
- Räkna på tid för sökning i en tabell
  - hög, ordnad, hash och indexerad
  - effekter på lagringsutrymmet
- Transaktioner
  - Lås - protokoll

Övning:

- En tabell lagras i en fil. Tabellen har 40 000 rader.
- Hur många poster har filen? r=
- Posterna i filen tar plats, storleken på en post är summan av datautrymmet för de attribut som lagras för varje rad. R=400 byte.
- Hur stor plats tar filen?
  1. Vi räknar i antal block. Kom ihåg att vi inte lagrar en del av en post i ett block. B= 4096 byte.
  2. Beräkna hur många poster som får plats i varje block. bfr = ⌊B/R⌋ = ⌊4096/400⌋

forts Övning

r = 40 000 st R=400 byte

B = 4096 byte bfr = 10

3. Beräkna hur många block hela filen tar upp:
   b = ⌈r/bfr⌉ = ⌈40 000/10⌉ = 4000 st block
4. Svar: Filen tar upp 4000 block.

Övning söktid

- Givet filen som tar upp b = 4000 block, hur lång tid tar det att söka en viss post i den?
- Beror på filorganisationen (hög, sorterad, hash eller indexerad).
- hög: söker igenomsnitt igenom halva filen: ⌈b/2⌉ block läses in från hårddisken. accesstid = 10 ms = 0,01 s
- antal blockaccesser: ⌈b/2⌉ = ⌈4000/2⌉ = 2000 st
- tid: antal * accesstid = 2000 * 0,01 = 20 sekunder

Övning söktid: ordnad

b = 4000 accesstid = 0,01 s

- Sök med hjälp av binärsökning
- antal block att läsa in $n = \lceil \log_2(b) \rceil$
  $n = \lceil \log_2(4000) \rceil = \lceil 11{,}966 \rceil = 12$ accesser
- i tid: 12 * 0,01 = 0,12 sekunder.

Övning söktid: hash

- Hashfunktionen sprider posterna över blocken
- sökning på nyckeln = beräkna värdet, hämta blocket
- Alltid 1 blockaccess
- accesstid 0,01 ger 0,01 sekund

- Men om spillblock: 2-3 accesser, 0,02-0,03 sekunder

Övning söktid: index

- Varje indexpost består av en nyckeln till en posten och adressen till ett block. Dessa tar viss plats.
- Primärindex: Datafilen är sorterad på primärnyckeln och ligger i en serie block. Indexposten innehåller nyckeln till första posten i första blocket och pekaren till det blocket, första posten i andra blocket och pekaren till det blocket osv. (Glest index)
- Sökning med hjälp av index: sök i indexfilen efter det block som omfattar det sökta värdet. Använd pekaren för att läsa in det blocket. Sökningen görs med binärsökning.

forts. övning söktid: index (primärindex)

- Index för filen: Antag att indexpostens storlek iR=40 byte. Datafilen har r block = 4000 st. Indexfilen kommer att ha ir= 4000 poster.
- Själva indexfilen är alltid sorterad. Sökning: beräkna indexfilens antal block (indexfilen antas lagras på samma hårddisk, dvs blockstorleken är densamma):
- ibfr = ⌊B/iR⌋ = ⌊4096/40⌋ = 102 poster per block.
- indexfilens antal block: ib = ⌈ir/ibfr⌉ = ⌈4000/102⌉ = ⌈39,21⌉ = 40st block

forts. övning söktid: index (primärindex)

- antal blocköverföringar n = $\lceil \log_2(40) \rceil = \lceil 5{,}32 \rceil = 6$ accesser. Plus 1 för accessen till datafilen = 7 accesser.
- i tid: 7 * 0,01 = 0,07 sekunder.

- Notera att indexfilen tar plats, för att lagra vårt data plus indexfilen krävs nu 4040 block.

forts. övning söktid: index (sekundärindex)

- Indexet görs av ett unikt attribut som datafilen INTE är sorterad på.
- Indexposten består av indexvärdet och pekaren till det block den posten finns i. Antalet poster i indexfilen är en per post i datafilen ir = 40 000.
- Hur lång tid tar det att söka med detta index?
- antal block i indexfilen = ib = ⌈ir/ibfr⌉ = ⌈40000/102⌉ = ⌈392,16⌉ = 392 st block
- antal accesser n = $\lceil \log_2(392) \rceil + 1 = \lceil 8{,}61 \rceil + 1 = 10$
- i tid: 10 * 0,01 = 0,1 sekunder.
- Datautrymme: 4000 + 392 block.

Transaktioner loggning

- Givet en loggfil, avgör vad som behöver göras när den läses igenom.

start (T1)
läs (T1, x)
skriv (T1, x, 300, 400)
läs (T1, y)
skriv (T1, y, 800, 700)
commit (T1)

start (T2)
läs (T2, x)
skriv (T2, x, 300, 400)
(slut på filen)

Svar: Alla T1s skriv ska göras om (repeteras). T2s skriv ska återställas (rollback).

Transaktioner - parallell-problem

- Givet två transaktioner, visa hur bortkastad uppdatering händer

T1:	T2
start	start
läs (x)	läs (x)
skriv (x)	skriv (x)
commit	commit

Svar:

Transaktioner lås

- Lägg in binära lås (lås/låsupp) i en transaktion så att den följer tvåfaslåsning