

732G I 6: Databaser; Design och programmering

Övning

Fysisk design av databasen

Transaktioner

Innehåll

1. Räkna på utrymme för en tabell
2. Räkna på tid för sökning i en tabell
 - 2.1.hög, ordnad, hash och indexerad
 - 2.2.effekter på lagringsutrymmet
3. Transaktioner, loggning
4. Lås - bortkastad uppdatering och protokoll

Övning 1: Räkna på utrymme för tabell

- En tabell lagras i en fil, en rad blir en post
- Tabellen har 40 000 rader. Hur många poster har filen? $r =$
- Cellerna på raden blir fält i posten. Storleken på en post är summan av datautrymmet för de attribut som lagras för varje rad.
- Poststorlek $R = 400$ byte.
- Hur stor plats tar filen??
 1. Hur räknar vi?

Övning 1 forts:

- Vi räknar i antal block på hårddisken!
 - Kom ihåg att vi inte lagrar en del av en post i ett block.
 - Hårddiskens blockstorlek $B = 4096$ byte.
1. Beräkna hur många poster som får plats i varje block (blockningsfaktor, bfr).
 2. $bfr = \lfloor B/R \rfloor$
 3. $bfr = \lfloor 4096/400 \rfloor =$

Övning 1 forts.

Vi har följande data hittils:

$r = 40\,000$ st $R = 400$ byte

$B = 4096$ byte $bfr = 10$

4. Beräkna hur många block hela filen tar upp:
5. $b = \lceil r/bfr \rceil$
6. $b = \lceil 40\,000/10 \rceil = 4000$ st block
7. Svar: Filen tar upp 4000 block.

Övning 2: Räkna på tid för sökning i en tabell

- Givet filen som tar upp $b = 4000$ block, hur lång tid tar det att söka en viss post i den?
- Beror på filorganisationen
 - Hög (osorterad fil)
 - Ordnad fil (sorterad)
 - Hashstruktur
 - Indexerad fil

Övning 2: Sökning i hög

- Hur söker vi?
 - Linjärt genom filen: söker i genomsnitt genom halva filen.
 1. antal block att läsa in (blockaccesser) = $\lceil b/2 \rceil$ block
 2. antal accesser = $\lceil 4000/2 \rceil = 2000$ st
 3. Vad blir det i tid? accesstid = 10 ms = 0,01 s
 4. tid = antal * accesstid = 2000 * 0,01 = 20 sekunder
- Svar: 2000 accesser, 20 sekunder.
- I värsta fall, eller om posten inte finns: b accesser

Övning 2: Sökning i ordnad fil

$b = 4000$ $\text{accesstid} = 0,01 \text{ s}$

- Hur söker vi?
- Med hjälp av binärsökning
 1. Antal block att läsa in
 2. $n = \lceil \log_2(b) \rceil$
 3. $n = \lceil \log_2(4000) \rceil = \lceil 11,966 \rceil = 12$ accesser
 4. Vad blir det i tid?
 5. $12 * 0,01 = 0,12$ sekunder.

Svar: 12 accesser, 0,12 sekunder

- Sökning på annat än filen är ordnad på: som i hög!

Övning 2: Sökning i hashstruktur

- Hur söker vi?
- Sökning på nyckeln: Hashfunktionen ger ”rätt” block direkt. Om spillblock, en eller två till.
 1. Beräkna hashvärdet, hämta blocket!
 2. Alltid 1 blockaccess, 2-3 om spillblock använts.
 3. Vad blir det i tid: accesstid 0,01 ger 0,01 sekund, eller 0,02-0,03 sekunder för spillblock

Svar: 1-3 accesser, 0,01-0,03 s

- Men sökning på annat än nyckeln: som i hög!

Övning 2: Sökning i Indexerad fil

- Hur söker vi?
- Sök i indexet först, ger adressen (pekaren) till rätt block. Läs in det.
- Hur fungerar sökningen i indexfilen?
- Som sökning i sorterad fil (indexet självt är alltid sorterat).
- Sökning i sorterad fil: $n = \lceil \log_2(b) \rceil$ accesser.
- Behöver räkna ut hur stor indexfilens b är!

Forts. övning 2: Sökning i Indexerad fil: Primärindex: Beräkna ib

- Hur stor plats tar indexfilen?
 - Varje indexpost innehåller nyckeln till en post och adressen (pekaren) till det block den posten (den raden) ligger i.
 - Primärindex: Datafilen är sorterad på primärnyckeln och tar b block. Här behövs en indexpost per block i datafilen (Glest index).
1. $ir = b$

Forts. övning 2: Sökning i Indexerad fil: Primärindex: Beräkna ib

- Antag att indexpostens storlek $iR=40$ byte. Datafilen har r block = 4000 st, och hårddisken blockstorleken 4096 byte
 1. Indexfilen kommer att ha $ir= 4000$ poster.
 2. $ibfr = \lfloor B/iR \rfloor =$
 3. $ibfr = \lfloor 4096/40 \rfloor = 102$ poster per block.
 4. indexfilens antal block: $ib =$
 5. $ib = \lceil ir/ibfr \rceil =$
 6. $ib = \lceil 4000/102 \rceil = \lceil 39,21 \rceil = 40$ st block

Forts. övning 2: Sökning i Indexerad fil: Primärindex: sökning i indexet:

- Indexfilen har $ib = 40$ block.
 1. Sökning i indexfilen (binärsökning)
 2. $n = \lceil \log_2(40) \rceil = \lceil 5,32 \rceil = 6$ accesser.
 3. Plus 1 för accessen till datafilen = 7 accesser.
 4. Vad blir det i tid: $7 * 0,01 = 0,07$ sekunder.

Svar: 7 accesser, 0,07 s

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

Forts. övning 2: Sökning i Indexerad fil: Sekundärindex: beräkna ib

- 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 $ir = 40\ 000$.
1. Hur stor bli indexfilen?
 2. antal block i indexfilen = $ib = \lceil ir/ibfr \rceil$
 3. $ib = \lceil 40000/102 \rceil = \lceil 392,16 \rceil = 393$ st block

Forts. övning 2: Sökning i Indexerad fil: Sekundärindex:

- $ib = 393$ block
- 1. Hur lång tid tar det att söka med detta index?
- 2. (Binärsökning $n = \lceil \log_2(n) \rceil$)
- 3. $n = \lceil \log_2(393) \rceil + 1 = \lceil 8,61 \rceil + 1 = 10$
- 4. Vad blir det i tid: $10 * 0,01 = 0,1$ sekunder.

Svar: 10 accesser, 0,1 s

- Datautrymme som krävs: $4000 + 393$ block.

Övning 3: Transaktioner loggning

- Givet en loggfil med följande innehåll, avgör vad som behöver göras när den läses igenom efter att hårddisken kraschat.

1. (filen startar här)
2. start (T1)
3. läs (T1, x)
4. skriv (T1, x, 300, 400)
5. läs (T1, y)
6. skriv (T1, y, 800, 700)
7. commit (T1)
8. start (T2)
9. läs (T2, x)
10. skriv (T2, x, 400, 500)
11. (slut på filen)

Övning 3 svar

- Hårddiskkrasch, läs tillbaka backupen först.
- Sedan: färdiga transaktioner repeteras (det nya värdet skrivs i db), Ofärdiga transaktioner rivs upp (det gamla värdet skrivs i db).

1. (filen startar här)
2. start (T1)
3. läs (T1, x)
4. skriv (T1, x, 300, 400)
5. läs (T1, y)
6. skriv (T1, y, 800, 700)
7. commit (T1)
8. start (T2)
9. läs (T2, x)
10. skriv (T2, x, 300, 400)
11. (slut på filen)

Svar: Nr 4 och 6 repeteras, 10 rivs upp.

Övning 4: Transaktioner - problem

- Givet två transaktioner enligt nedan, visa hur bortkastad uppdatering händer

T1:

start

läs (x)

skriv (x)

commit

T2

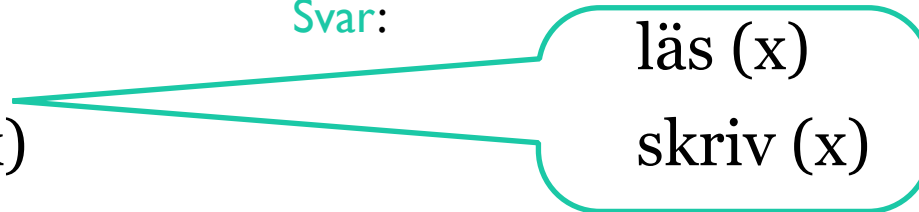
start

läs (x)

skriv (x)

commit

Svar:



Övning 4: Transaktioner - lås

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

Svar:

