

Tentamen 2003-05-20 DATABASTEKNIK - 1DL116

DatumTisdagen den 20 Maj, 2003
Tid12:00-17:00
Jourhavande lärare ...Kjell Orsborn, tel. 471 11 54 eller 070 425 06 91
HjälpmedelInga

Anvisningar:

- Läs igenom hela skrivningen och notera eventuella oklarheter innan du börjar lösa uppgifterna. Förutom anvisningarna på skrivningsomslaget så gäller följande:
 - Skriv tydligt och klart. Lösningar som inte går att läsa kan naturligtvis inte ge några poäng och oklara formuleringar kan dessutom misstolkas.
 - Antaganden utöver de som står i uppgiften måste anges. Gjorda antaganden får förstås inte förändra den givna uppgiften.
 - Skriv endast på en sida av papperet och använd ett nytt papper för varje uppgift för att underlätta rättning och minska risken för missförstånd.
- För godkänt krävs det cirka 50% av maxpoäng.

1. DB terminologi:

4p

Förklara kortfattade följande begrepp i databassammanhang (det går bra med definitioner där sådana finns).

- (a) referensnyckel (eng. foreign key)
- (b) transaktion
- (c) dödlig låsning (eng. dead lock)
- (d) BCNF - Boyce-Codd Normal Form

Svar:

- (a) En referensnyckel är en mängd av attribut (rk) i relationsschema R1 som refererar till en mängd attribut av samma värdeomän och som samtidigt utgör primärnyckel (pk) i relationsschema R2. Värdet av rk för tuple t1 i en relation r1(R1) är detsamma som värdet av pk för någon tuple t2 i en relation r2(R2), eller NULL.
- (b) En databastransaktion är en atomisk och logisk enhet av databas processering som accessar och eventuellt uppdaterar olika data items. En transaktion genomförs alltid antingen i sin helhet eller inte alls (vilket garanteras av transaktionshanteraren som ser till att transaktioner hanteras som en odelbar mängd av operationer).
- (c) Dödlig låsning (eng. dead lock) är en situation som kan uppstå när alla transaktioner i en mängd av två eller flera transaktioner väntar på att få accessrättinghet till någon dataartikel som är låst av någon annan av transaktionerna.
- (d) BCNF - Boyce-Codd Normal Form säger att ett relationsschema uppfyller BCNF om det uppfyller 1NF samt att varje determinant (dvs vänsterled i ett funktionellt beroende) skall vara en kandidatnyckel.

2. Databassystem - konventionell filprocessering:

4p

Nämn och motivera fyra fördelaktiga egenskaper hos ett databassystem som normalt skiljer sig från egenskaperna hos ett programsystem som baseras på konventionell filprocessering.

Svar: Exempelvis: Metadata ..., Dataoberoende ..., Vyer ..., Datadelning och fleranvändar-transaktionshantering ..., Kontroll av redundans ..., Begränsning av otillbörlig access ..., Persistens ..., Aktiva regler ..., Multipla gränssnitt ..., Representation av komplexa datasamband ..., Integritetskontroll ..., Backup och recovery ...

3. Datamodeller:

4p

- (a) Hur representerar man begreppen entitetstyp (eng. entity type) och attribut (eng. attribute) från en E-R modell i relationsmodell? (1p)

- (b) Hur kan man representera binära relationstyper i relationsmodellen (notera att det finns flera binära relationstyper)? (3p)

Svar:

- (a) E-R modellens begrepp entitetstyp representeras som en tabell (relation) i relationsmodellen och attribut representeras som en namngiven kolumn (attribut) i en tabell (relation) (1p)
- (b) En 1-1 relationstyp mellan två entitetstyper representeras genom att inkludera primärnyckeln från tabellen som representerar den ena entitetstypen som en främmande nyckel i tabellen för den andra entitetstypen; en 1-N relationstyp mellan två entitetstyper representeras genom att inkludera 1-sidans primärnyckeln som en främmande nyckel på N-sidan; en M-N relationstyp mellan två entitetstyper representeras som en egen tabell (relation) med två främmande nycklar (3p)

4. Fysisk databasdesign - indexering:

4p

Förklara för vilka ändamål (typer av databasfrågor) som följande index kan, och inte kan, effektivisera exekveringen:

- (a) hashindex
(b) B^+ -träd

Svar:

- (a) Hashindex är effektiva för sökning av godtyckliga poster med avseende på värdet av hashfältet. Hashindex är mindre lämpliga (kan jämföras med sökning i ordnad fil) för att söka efter värden med avseende på något annat fält än indexeringsfältet. De är normalt heller ej lämpliga för sökning av ordnade poster då det kan krävas en diskaccess för varje post.
- (b) B^+ -träd är effektiva för sökning av poster i ordning baserad på indexeringsfältet och för frågor som inbegriper sökvillkor baserat på indexeringsfältet. Exempelvis villkor som inbegriper $<$, $>$, \leq , och \geq betyder att posterna som uppfyller villkoret lagras kontinuerligt efter varann. Frågor som innebär access av godtyckliga poster eller av poster ordnade efter något annat fält än indexeringsfältet ges inga speciella fördelar av ett trädindex.

5. Aktiva databaser:

4p

- (a) Vad består komponenterna i ECA-modellen av i en relationsdatabas? (3 p)

- (b) Varför är det ofta svårt att implementera komplexa integritetsvillkor m.h.a. triggers? (1 p)

Svar:

- (a) E = Event = Uppdatering av tabellrad (insert, delete, update)
C = Condition = Databasfråga som ger icke-tomt svar om triggeren skall utlösas
A = Action = en eller flera uppdateringar eller abort (3 p)
- (b) Det är svårt att identifiera alla möjliga situationer och triggerutlösningar kan ofta leda till svåröverskådliga kedjereaktioner (1 p)

6. Objekt-orienterade databaser: 4p

- (a) Vilka tre sorters användardefinierade databasutvidgningsmekanismer finns i ett objekt-relacionellt databashanteringssystem?
- (b) Vilka av ovanstående utvidgningsmekanismer saknas eller är svaga i enkla objekt-orienterade databashanterare (dvs. i s.k. object stores)?

Svar:

6a. Användardefinierade datatyper Användardefinierade index Användardefinierad frågeoptimering

6b. Det saknas stöd för användardefinierad frågeoptimering Det saknas stöd för användardefinierade index

7. Databasapplikationsgränssnitt: 4p

- (a) Vad är JDBC? (1 p)
- (b) Beskriv JDBC:s arkitektur med text och bild. (2 p)
- (c) Vad är skillnaden mellan JDBC och ODBC? (1 p)

Svar:

7a. Ett standardgränssnitt mellan Java och en eller flera samtidigt tillgängliga relationsdatabaser

7b. Se kompendium

7c. ODBC är programmeringsspråks- och plattformsoberoende.

8. Frågeoptimering: 4p

En stor firma har en relationsdatabastabell över hur bra deras försäljare är:

SALES(PNR, SALES, NAME, ...)

Tabellen innehåller försäljning (SALES) i kr för varje försäljare med nyckel personnummer (PNR). Det finns klustrat primärindex på PNR och oklustrat sekundärindex (B-träd) på SALES. Det finns 10000 försäljare i tabellen. Det får plats 10 tabellrader och 100 indexnoder i varje diskblock.

Företagsledningen behöver ofta veta de 10 bästa försäljarna och du ombeds att designa ett applikationsprogram som snabbt finner dessa stjärnor. För sådana frågor tillhandahåller SQL nuförtiden en utvidgad syntax:

```
select ssn, sales
stop after 10 rows
from sales
order by sales descending
```

Klausulen 'stop after 10 rows' betyder att bara de 10 första raderna i frågan kommer att returneras.

Vilken är den optimala exekveringsplanen uttryckt som utvidgad relationsalgebra innehållande stopafter(x,n) operator? Visa varför den är optimal. Vilken betydelse har 'stop after 10 rows' klausulen för effektiviteten av exekveringsplanen?

Svar:

8. Två huvudalternativ:

1: stopafter(indexscandescending(SALES.SALES),10)

2: stopafter(sort(segmentscan(SALES), 10)

3: topsort(segmentscan(SALES), 10)

1. traverserar indexet från slutet sekevntiellt och plockar upp raderna. Ingen sortering behövs. 12 diskaccesser. (2 indexnoder + 10 radnoder)

2. Söker igenom hela filen 5 ggr, dvs 5000 diskaccesser för sorteringen.

3. Söker igenom hela filen 1 gång och håller 10 största i primärminnet. 10000 diskaccesser.

Plan 1 är optimal. Utan stop after 10 kan inte optimeraren veta att sökningen skall avbrytas efter 10 första raderna. plan 2 kan då bli effektivare eftersom det går åt ca 10000 diskaccesser att hämta alla tabellraderna via det oklustrade indexet på SALES.SALES.

Lycka till!

/ Kjell och Tore