

Tentamen 2004-08-23

DATABASTEKNIK - 1DL116, 1MB025, 1DL124

DatumTisdagen den 23 Augusti, 2005
Tid8:00-13:00
Jourhavande lärare ...Kjell Orsborn, tel. 471 11 54 eller 070 425 06 91
Hjälpmedelminiräknare

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. Databasterminologi:

4 p

Förklara följande databasbegrepp:

- (a) referensintegritet (eng. referential integrity)

Svar: Referensintegritet kräver att om en tupel i en relation refererar till en annan relation så måste den referera till en existerande tupel.

- (b) entitetsintegritet (eng. entity integrity)

Svar: Entitetsintegritet uttrycker att ingen primärnyckel får anta värdet NULL så att alla tupler i en relation kan identifieras unikt.

- (c) supernyckel (eng. super key)

Svar: En supernyckel är varje delmängd av en relations attribut som unikt kan identifiera alla tupler i relationen (notera att det normalt finns fler än en supernyckel för samma relation).

- (d) naturlig join (eng. natural join)

Svar: En naturlig join är en sammansättningsoperation mellan två relationer (tabeller) där villkoret för att kombinationen av två tupler (en från varje relation) skall ingå i den resulterande och sammansatta relationen är ett likhetsvillkor mellan ett/ flera attribut. De attribut från den högra relationen som deltar i likhetsvillkoret ingår ej i den resulterande tabellen, dvs redundanta attribut elimineras.

2. Datamodeller:

4 p

I utvidgad entitets-relationsmodellering (eng. enhanced entity-relationship, EER, modeling), stöds gruppering av entiteter i olika grad. Förklara vilken information de två följande mekanismerna representerar och hur de kan representeras i EER:

- (a) generalisering (eng. generalization) (2p)

Svar: Generalisering är processen att specificera en superklass utifrån ett antal subclasser där man finner ett antal gemensamma egenskaper som kan extraheras och defineras att utgöra egenskaperna i en gemensam superklass. Dessa egenskaper ärvs sedan till subclasserna.

- (b) aggregering (eng. aggregation) (2p)

Svar: Aggregation is an abstraction concept to group entities into composite objects from their components. In three cases can aggregation be related to the EER model. The 1st case is an aggregation of attribute values of an object to form the whole object. The 2nd case is the representation of an aggregation relationship using an ordinary relationship. The 3rd case is not explicitly supported in EER but involve the possibility to combine related objects using a particular relationship instance into a higher-level aggregate object.

3. SQL:

4 p

Anta att vi har en databas för byggsatser med två relationer (tabeller) med följande scheman:

```
BYGGSATS(BID, BNAMN)
DEL(DID, DNAMN, PRIS, FÄRG, BGID)
```

, där xID's representerar nycklar.

- (a) Formulera en fråga i relationsalgebra som återfinner byggsatsens id och namn samt delens id, namn och pris för byggsatsen "Starship Enterprise". (2p)
- (b) Formulera en SQL fråga som återfinner id, namn, och antal delar för varje byggsats (alltså hur många delar som varje byggsats består av). (2p)

Svar: $\pi_{\langle BID, BNAMN, DID, DNAME, FARG \rangle}$

$(\sigma_{BNAME='StarshipEnterprise'}(BYGGSATSX_{\langle PID=BGID \rangle DEL}))$

```
SELECT B.BID, B.BNAMN, COUNT(*) AS ANTAL_DELAR
FROM BYGGSATS B, DEL D
WHERE B.BID = D.DID
GROUP BY BID, BNAME
```

4. Fysisk databasdesign:

4 p

Beskriv kortfattat organisationen av och funktionen för hash-filer (svaret skall bland annat innefatta hur man återfinner (söker) en datapost för en specifik söknyckel där datafilen är en hash-fil)

Svar: En hash-fil består av ett statiskt eller dynamiskt antal datablock som hanteras av olika typer av hashningstekniker. Hash-filer hanterar adressering av dataposter till datablock genom att applicera en hash-funktion till hash-fältet (dvs sökfältet) vilken returnerar adressen till ett datablock för insättning eller återsökning av dataposten. En vanlig form av hashfunktion har formen $h(f(p)) = f(p) \bmod M$, där hash-funktionen $h(f(p))$ tillhandahåller adressen för det datablock där dataposten p skall lagras genom att beräkna hashfältet $f(p)$ modulo (mod) antalet datablock M . Man hittar alltså var (i vilket block) en datapost finns för en specifik söknyckel genom att beräkna hash-funktionen för nyckeln som ger adressen till blocket.

5. Samtidighetskontroll (eng. concurrency control):

4p

Beskriv principerna för låshanteringen i samband med transaktioner för ett två-fas låsningsprotokoll (eng. two-phase locking protocol).

Svar: En transaktion sägs följa ett två-fas låsningsprotokoll om alla låsningsoperationer föregår den första upplåsningsoperationen (unlock) i transaktionen. Alltså en sådan transaktion genomgår en expanderande fas där nya lås kan utfärdas men inga lås kan släppas; och en krympande fas där existerande lås kan låsas upp men inga nya lås kan erhållas.

Två-fas låsningsprotokoll garanterar serialiserbara transaktionsscheman.

6. Databasapplikationsgränssnitt:

4 p

- (a) Vad är JDBC? (1 p)

Svar: JDBC = Java DataBase Connectivity är ett standardgränssnitt mellan programmeringsspråket Java och en eller flera samtidigt tillgängliga SQL-baserade relationella databaser.

- (b) Vad är skillnaden mellan JDBC och ODBC? (1 p)
 Svar: ODBC = Open DataBase Connectivity är ett programmeringsspråks- och plattformsoberoende gränssnitt till SQL-baserade relationella databaser.
- (c) Vad står "O" för i "ODBC"? (1 p)
 Svar: O står för open och syftar på programmeringsspråksberoende och operativsystemberoende.
- (d) På vilket sätt tillåter JDBC frågor som returnerar mycket stora datamängder? (1 p)
 Svar: JDBC hanterar frågeresultat som resultatströmmar genom ett ResultSet objekt som representerar en resultattabell där en rad i taget kan genereras och bearbetas genom att stega sig igenom resultattabellen.

7. Frågeoptimering:

4 p

- (a) Vad kallas de tre viktigaste join algoritmerna och hur mycket minne behöver de när de körs? (3p)
 Svar:
 Sort-merge join: I princip behövs endast minne för att hantera två poster samtidigt för jämförelse.
 Nested-loop join: I princip behövs endast minne för att hantera två poster samtidigt för jämförelse.
 I praktiken så läses flera poster in samtidigt från båda tabellerna i varsin buffert och dessutom finns en resultatbuffert. Alltså behövs minnesutrymme för tre minnesbuffertar.
 Hash join: Här behövs minnesutrymme för en hashtabell över ena tabellen samt för att hantera en post för den andra tabellen. Även här hanteras i praktiken buffertar med flera poster samtidigt för data- och resultatströmmar.
- (b) Hur hanterar "embedded SQL" den höga kostnaden att optimera SQL-frågor (1p)
 Svar: Inbäddade SQL-frågor i värdspråket extraheras av en prekompilator och förkompileras vid kompileringen av applikationen. Den kompilerade frågan ersätts av motsvarande databasanrop som länkas samman med applikationen och vid exekveringen sker databasanrop från applikationen. Vid senare (eventuellt upprepad) exekvering behövs ej någon kompilering vilket eliminerar onödig frågeoptimering.

8. Datalager:

4 p

Ett konsult behöver analysera sin verksamhet och tänker därför utnyttja datalagerteknik. Man vill analysera uppdragens inkomster per kund och typ av uppdrag (t.ex. seminarium, design, implementering, testning).

- (a) Hur ser datakuben ut som sammanfattar ovanstående? Ge exempel. (1 p)
 Svar:

Uppdrag \ Kund	abb	sca	s:a
seminarium	100	90	190
design	75	80	155
implementering	125	120	245

```

testning      | 50 | 65 | 115
-----
s:a           | 350 | 355 | 705

```

- (b) Designa ett stjärnschema för att lagra datakuben i en relationsdatabas. Ge exempel på tabellinnehåll. (2 p)

Svar:

```

Kund(kundid, namn, + andra egenskaper)
Uppdrag(uppdragid, namn, + andra egenskaper)
Konsultinkomster(kundid, uppdragid, inkomst)

```

- (c) Hur uttrycker man fråga i SQL m.h.a. cube-operatorn för att konstruera datakuben från relationsdatabasen? Hur ser resultattabellen ut? (1 p)

Svar:

```

select kundid, k.namn as knamn, uppdragid, u.namn as unamn,
       sum(inkomst)
from Kund k, Uppdrag u, Konsultinkomster ki
where k.kundid = ki.kundid and
       u.uppdragid = ki.uppdragid
group by kundid, k.namn, uppdragid, u.namn
with cube;

```

resultattabell:

```

kundid | knamn | uppdragid | unamn          | inkomst
-----
k1     | abb  | u1        | seminarium    | 100
k1     | abb  | u2        | design        | 75
k1     | abb  | u3        | implementering | 125
k1     | abb  | u4        | testning      | 50
k2     | sca  | u1        | seminarium    | 90
k2     | sca  | u2        | design        | 80
k2     | sca  | u3        | implementering | 120
k2     | sca  | u4        | testning      | 65
null   | null | u1        | seminarium    | 190
null   | null | u2        | design        | 155
null   | null | u3        | implementering | 245
null   | null | u4        | testning      | 115
k1     | abb  | null     | null          | 350
k2     | sca  | null     | null          | 355
null   | null | null     | null          | 705
-----

```

Lycka till!

/ Kjell och Tore