Uppsala universitet
Institutionen för informationsteknologi
Kjell Orsborn, Tore Risch

Tentamen 2004-05-30
DATABASTEKNIK - 1DL116, 1MB025

```
Datum .....................................Tisdagen den 30 Maj, 2005
Tid ...........................................................14:00-19:00
Jourhavande lärare ...Kjell Orsborn, tel. 471 11 54 eller 070 425 06 91

Hjälpmedel ..............................................miniräknare
```

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)

(b) transaktion

(c) objektidentifierare (OID)

(d) Boyce Codds normalform (BCNF)

Svar:

(a) Referensintegritet kräver att om en tupel i en relation refererar till en annan relation så måste den referera till en existerande tupel.

(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) en objektidentifierare (OID) är en unik och ofta logisk och systemgenererad identifierare som används för att unikt identifiera objekt under hela dess existens, samt för att hantera referenser mellan objekt.

(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. **Datamodeller:** 4 p

(a) Beskriv relationsdatamodellen där beskrivningen skall omfatta begreppen relationschema, rad, nyckel, kolumn, värdedomän. (3p)

Svar: Relationsdatamodellen representerar en databas som en samling relationer (eller tabeller). Varje tabell har ett namn och representerar ett fysiskt eller abstrakt begrepp eller samband. Begreppets eller sambandets egenskaper representeras av tabellens kolumner (eller attribut) med kolumnens namn och värdedomän. Värdedomänen anger vilka tillåtna värden som attributet kan ha.

Varje rad (eller tupel) i tabellen representerar en specifik individ av begreppet eller sambandet och omfattar en mängd av samhörande värden, ett värde för varje attribut i tabellen. Varje rad i tabellen är vidare unik och särskiljs av att ett eller flera attribut har unika värden för varje rad. Detta (eller dessa) attribut sägs utgöra tabellens nyckel och används för att unikt identifiera varje rad i en tabell. En tabell omfattar alltså en mängd av rader där varje rad representerar ett individuellt begrepp eller samband.

Ett relationsschema beskriver en tabells gemensamma struktur i forma av relationens/tabellens namn och dess gemensamma mängd av attribut. Ordningen mellan attribut eller mellan tupler har ingen betydelse i relationsmodellen.

(b) Vad är 1:a normalformen för relationsdatamodellen? (1p)

Svar: Första normalformen säger att alla värden i en relation/tabell endast tillåts vara atomisk. Alltså varje värde skall betraktas som odelbart så att sammansatta eller multipla värden ej är tillåtna.

3. **SQL:** 4 p

Anta att vi har en produktdatabas med två relationer (tabeller) med följande scheman:

```
PRODUKT(PID, PNAMN)
DETALJ(DID, DNAMN, PRIS, FÄRG, PID)
```

, där xID's representerar nycklar.

(a) Formulera en fråga i relationsalgebra som återfinner produktid, produktnamn, detaljid, detaljnamn och detaljernas färg för produkten "Evighetsmaskin". (2p)

(b) Formulera en SQL fråga som återfinner de produktid, produktnamn, och antal delar (part) för varje produkt (alltså hur många delar som varje produkt består av). (2p)

Svar: $\pi_{<PID,PNAME,PRTID,PRTNAME,COLOUR>}$
$(\sigma_{PNAME='Evighetsmaskin'}(PRODUCT \texttt{X}_{<PID=PRODID>} PART))$

```
SELECT P.PID,P.PNAME, COUNT(*) AS NO_OF_PARTS
FROM PRODUCT P, PART C
WHERE P.PID = C.PID
GROUP BY PID,PNAME
```

4. **Fysisk databasdesign - indexering:** 4 p

Förklara för vilka 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 oordnad 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 $<$, $>$, $\leqslant$, och $\geqslant$ 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. **Återhämtning (eng. recovery):** 4 p

Beskriv kortfattat proceduren för återhämtning enligt modellen omedelbar uppdatering (eng. immediate update) i en fleranvändarversion och där strikta transaktionsplaner antas.

Svar:

Recovery according to the immediate update model:

1. Start from the last record in the log file and traverse backwards until a check point is reached. Create two lists: C transactions that have reached their commit points NC transactions that have not reached their commit points.

2. Start from the last record in the log file and apply the UNDO procedure to all (Write,T,...) where $T \in NC$.

3. Start from the check point and REDO all transactions (Write,T,...) such that $T \in C$.

4. Restart all failured transactions.

6. **Aktiva databaser:** 4 p

(a) Vad består komponenterna i ECA av i en relationsdatabastrigger? (3 p)

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

Svar:

6a: E: Event, uppdatering av rad i tabell

C: Condition, databasfråga som måste vara icke-tom för att triggern skall utlösas.

A: Action, databasupdateringar aller anrop till databasprocedur

6b: Man måste ta hänsyn till alla möjliga situationer. T.ex. kan ett integritetsvillkor påverkas av många olika typer av uppdateringar och triggers måste då defineras för alla dessa.

7. **Frågeoptimering:** 4 p

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 ett klustrat primärindex på PNR och ett 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 program som snabbt finner dessa stjärnor. För sådana frågor tillhandahåller SQL nuförtiden en utvidgad syntax (varierar beroende på system):

```
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.

(a) Vilken är den optimala exekveringsplanen uttryckt i utvidgad relationsalgebra som har vanliga relationsalgebraoperatorer + segment_scan, index_scan och sort. Dessutom finns operator stopafter(x,n) som läser första n raderna i strömmen s. Visa varför den valda planen är optimal. (3 p)

(b) Vilken betydelse har "stop after 10 rows"klausulen för val av exekveringsplan? (1 p)

Svar:

7a: Optimal plan:

$project(< ssn, sales >, stopafter(index_scan(sales.sales, reverse), 10))$

Kostnad: 10 blockläsningar. Läsning av 10 block från grundtabellen (1 per indexnyckel) genom det oklustrade indexet + 1 bock från indexet.

Alternativ plan:

$project(< ssn, sales >, sort(salesdescending, segment_scan(sales)))$

Kostnad: 5000 blockläsningar. Läsning av 1000 block i grundtabellen, Läsning av 4000 block för sortering (4 pass, 10 rader per block, 10**4)

7b) Om systemet inte vet att det skall stoppa efter 10 första indexraderna blir kostnaden för plan 1 mer än 10000 blockläsningar (oklustrat index) vilket är sämre än plan b.

8. **Datalager:** 4 p

Ett reseföretag behöver analysera sin verksamhet och tänker därför utnyttja datalagerteknik. Man vill analysera verksamheten som en datakub med försäljning av resor per kvartal och typ av resa (flyg, tåg, båt).

(a) Hur ser datakuben ut som sammanfattar ovanstående? Ge exempel. (2 p)

(b) Designa ett stjärnschema för att lagra datakuben i en relationsdatabas. (2 p)

Svar 8a:

```
Typ \ Kvartal | 1 | 2 | 3 | 4 | s:a
---------------------------------
flyg          | 10| 5 | 13| 22| 50
t|g           | 2 | 8 | 9 | 11| 30
b~t           | 7 | 2 | 5 | 8 | 22
---------------------------------
S:a           | 19| 15| 27| 41| 102
```

8b:

typ(tid, typnamn, andra egenskaper), tid är nyckel, dimensionstabell

kvartal(kvid, andra egeneskaper), kvid är nyckel, dimensionstabell

försäljning(tid,kvid,total), tid+kvid är nyckel, faktatabell

Lycka till och ha en solig sommar!

/ Kjell och Tore