

Omtentamen i DATABASER

Svar:

Obs! Lärare-version, med lösningar

DAG: lö, 28 nov 1998 **TID:** kl. 8.45 – 12.45 **SAL:** MG

Ansvarig: Martin Weichert
Förfrågningar: Martin Weichert, tel. 772 10 68
Resultat: anslås den 17 dec 1998
Poängantal: sammanlagt maximalt 60 poäng.
Betygsgränser: CTH: 3:a 24 p., 4:a 36 p., 5:a 48 p.
GU: Godkänd 28 p., Väl godkänd 48 p.
Hjälpmedel: utdraget ur *Oracle7 Server SQL Language*
resp. *Appendix B Syntax* ur referensmanualen för *Oracle SQL*
samt *SQL*Plus Quick Reference*.

Observera:

- Skriv tydligt och disponera pappret på ett lämpligt sätt.
- Börja varje uppgift på nytt blad. Skriv endast på en sida av pappret.
- Alla svar skall väl motiveras!
- Ange på tentan om du går på GU eller CTH. För CTH ange även vilken linje du går!

Lycka till!

Uppgift 1. (a) Ange tre orsaker hur felaktigheter i en databas kan uppstå. Till varje av dem, ange en lämplig metod som används som åtgärd mot dessa. (5p.)

Svar:

- Redundans, motsägande data på olika ställen → Normaliseringar
- Nonsens-data, t ex "31 juni" → typvillkor, undertypvillkor, inklusionsvillkor
- Tomma referenser → referensvillkor
- Ofullständiga ändringar → transaktionskonceptet
- Datorkrasch → loggfil, återställningsmetoder (fördröjd/omedelbar ändring)
- Parallellism, flera användare samtidigt → låsning (2-fas-, konservativ, strikt, ...), serialiserbarhet
- Obehörigt tillträde → säkerhetsmekanismer ("discretionary", "mandatory")

– 5p.

(b) För vilket syfte använder man **quorum-konsensus**-metoden? Förklara kort hur den fungerar. (3p.)

Svar: Beskrevs i föreläsning F12. – 3p.

8 poäng.

Uppgift 2. Låt $r(A, B, C)$ och $s(B, D)$ vara tabeller med följande innehåll:

r :

A	B	C
a_1	b_1	c_2
a_1	b_2	c_1
a_3	b_1	c_3

s :

B	D
b_1	d_1
b_1	d_2
b_2	d_3

Beräkna resultatet av följande relationsalgebra-uttryck!

- (a) $r \bowtie s$ (naturlig samkörning)
 (b) $\pi_A(r) \times \pi_D(\sigma_{B=b_1}(s))$

(Beakta att resultatet är **mängder**!)

Svar:

- (a) $r \bowtie s$: **2p.**
 (b) $\pi_A(r) \times \pi_D(\sigma_{B=b_1}(s))$: **3p.**

$r \bowtie s$:

A	B	C	D
a_1	b_1	c_2	d_1
a_1	b_1	c_2	d_2
a_3	b_1	c_3	d_1
a_3	b_1	c_3	d_2
a_1	b_2	c_1	d_3

$\pi_A(r) \times \pi_D(\sigma_{B=b_1}(s))$:

A	D
a_1	d_1
a_1	d_2
a_3	d_1
a_3	d_2

Kolumnerna i (a) får vara i annan ordning.

Poängavdrag (– 1p.) om det finns dubletter.

5 poäng.

Uppgift 3. En hyresgästförening vill hålla reda på fastighetsbolaget *Monopol* och använder en databas med följande tabeller:

- *fastighet(Gata, GNr, Byggd)*
En lista över hyreshus, med *Gata* och gatunummer *GNr*, som tillsammans unikt identifierar varje hus, samt året *Byggd* när huset byggdes.
- *lägenhet(LNr, Gata, GNr, Typ, Storlek, Hyra)*
En lista över alla lägenheter, med ett unikt lägenhetsnummer *LNr*, adressen som består av *Gata* och *GNr*, som tillsammans utgör en referens till *fastighet*, lägenhetens *Typ* ("etta", "tvåa" eller dyl.) och *Storlek* (yta i m²) och månadshyran *Hyra*.
- *bor(Person, LNr, Sedan)*
Lista över hyresgäster som bor i dessa lägenheter. *LNr* är referens till *lägenhet*, och *Sedan* är året sedan *Person* har bott i *LNr*.
- *reparation(Fel, LNr, År, Pris)*
En lista över reparationer som har åtgärdats eller skall åtgärdas i lägenheterna. För varje reparationsåtgärd registreras det vilket *Fel* och vilken lägenhet *LNr* (referens till *lägenhet*) det handlar om, samt vilket *År* reparationen har genomförts eller planeras att genomföras och till vilket *Pris*.

Skriv SQL-satser för följande uppgifter:

- (a) Lista alla personer som bor i någon etta på Hornsgatan.

Svar:

```
select PERSON
  from BOR B, LÄGENHET L
 where B.LNr = L.LNr
       and Typ = 'etta' and Gata = 'Hornsgatan';
```

eller

```
select PERSON
  from BOR
 where LNr in (select LNr
               from LÄGENHET L
               where Typ = 'etta' and Gata = 'Hornsgatan');
```

Obs – villkoret B.LNr = L.LNr får inte glömmas!

Tabellen FASTIGHET behövs inte! – **2p.**

- (b) Ange alla fastigheter på Valhallavägen där några reparationer gjordes eller planeras sedan året *efter* att huset byggdes (*d.v.s. året efter att det byggdes och alla efterföljande år*).

Svar:

```
select F.Gata, F.GNr
  from FASTIGHET F, REPARATION R, LÄGENHET L
 where R.LNr = L.LNr
       and L.Gata = F.Gata and L.GNr = F.GNr
       and L.Gata = 'Valhallavägen' and R.År > F.Byggd;
```

eller varianter med nästlade SELECT-satser.

Obs! Inga samkörningsvillkor får glömmas! – **2p.**

- (c) Lista lägenhetsnummer och kvadratmeterhyra på alla lägenheter på gatan Diplomatstaden som är större än alla lägenheter på Odengatan.

Svar:

```
select LNr, Hyra/Storlek
  from LÄGENHET
 where Gata = 'Diplomatstaden'
    and Storlek > ( select max(Storlek)
                   from LÄGENHET
                   where Gata = 'Odengatan' );
```

eller

```
... and Storlek > all ( select Storlek from ... )
```

– 3p.

- (d) Ange det sammanlagda priset på alla reparationer för lägenheter på Narvavägen under tiden 1980 – 1984.

Svar:

```
select sum(PRIS)
  from REPARATION R, LÄGENHET L
 where R.LNr = L.LNr
    and Gata = 'Narvavägen' and ÅR between 1980 and 1984;
```

– 2p.

- (e) Skapa en vy *statistik*(*Person_i_Läg*, *Antal_Läg*) som anger hur många lägenheter som finns med 1 person i, hur många med 2 personer i, osv. (Rader som innehåller noll behöver inte tas med.)

Skriv sedan ut denna vys innehåll i fallande ordning, d.v.s. sorterad från största till minsta antal personer.

Svar:

Endast relationen BOR behövs. Den måste grupperas efter LNr, och resultatet måste grupperas igen efter antalet personer. (Man **behöver** alltså 2 ggr. GROUP BY!) Har man SQL-92 så kan man skriva mellanresultatet som en SELECT-sats i FROM-delen av en annan SELECT-sats:

```
create view STATISTIK
  as
select PERSON_I_LÄG, count(*) as ANTAL_LÄG
  from ( select LNr, count(*) as PERSON_I_LÄG
        from BOR
        group by LNr )
group by PERSON_I_LÄG;
```

```
select * from STATISTIK
order by PERSON_I_LÄG desc;
```

Har man inte SQL-92, så måste man skapa en hjälpvy för mellanresultatet.

ORDER BY kan inte finnas i vy-definitionen! – 4p.

- (f) Enligt uppgifter från fastighetsbolaget kommer alla reparationer för lägenhet nummer L123 som planerades för 1998 eller senare att bli 50% dyrare och skjutas upp i ett år. Skriv in dessa ändringar i databasen!

Svar:

```
update REPARATION
  set PRIS = PRIS * 1.5
      ÅR = ÅR + 1
  where LNr = 'L123' and ÅR >= 1998;
```

– 2p.

15 poäng.

Uppgift 4. En bankkund har tre konton vid en bank. I början finns det 2000:- på konto A, 1000:- på konto B och 1000:- på konto C. Kunden vill transferera 200:- från konto C till konto B. För detta ska följande transaktion T1 genomföras:

T1: C:=C-200; – (minska konto C:s innehåll med 200)

T1: B:=B+200; – (öka konto B:s innehåll med 200)

Ungefär samtidigt börjar en transaktion T2 för ränteberäkning som ska höja alla kontobelopp med 10 procent:

T2: A:=A*1.1; – (öka konto A:s innehåll med 10%)

T2: B:=B*1.1; – (öka konto B:s innehåll med 10%)

T2: C:=C*1.1; – (öka konto C:s innehåll med 10%)

När de två transaktionerna genomförs så kan deras operationer sammanflätas på olika sätt. Betrakta följande två operationsföljder:

	T2: A:=A*1.1;	T1: C:=C-200;
	T1: C:=C-200;	T2: A:=A*1.1;
OF1:	T2: B:=B*1.1;	OF2: T1: B:=B+200;
	T2: C:=C*1.1;	T2: B:=B*1.1;
	T1: B:=B+200;	T2: C:=C*1.1;

- (a) Är operationsföljd OF1 serialiserbar? Om ja, vilken logisk följd av transaktionerna motsvarar den? Vilket blir slutresultatet?

Svar: Nej, den är **icke** serialiserbar, eftersom det finns krockande operationer som kräver olika logisk följd. (C- kommer före C* och kräver "T1 före T2", medan B+ kommer efter B* och kräver "T1 efter T2").

Slutresultatet blir: A=2200, B=1300, C=880; summa 4380 kronor. – 2p.

- (b) Är operationsföljd OF2 serialiserbar? Om ja, vilken logisk följd av transaktionerna motsvarar den? Vilket blir slutresultatet?

Svar: Ja, den är serialiserbar och motsvarar den logiska följden "T1 före T2" (alla operationer av T1 som **krockar** med några av T2 kommer före dessa: C- före C* och B+ före B*).

Slutresultatet blir: A=2200, B=1320, C=880; summa 4400 kronor. – 2p.

- (c) Hur mycket pengar förlorar kunden till banken vid den felaktiga operationsföljden?

Svar: 4400 - 4380 = 20 kronor. – 1p.

(d) Vilka av de följande metoderna kan garantera korrekt genomföring av transaktionerna?
Med vilken av dessa metoder kan ett annat problem uppstå? – Vilket? Motivera!

- låsning: varje dataobjekt (A, B och C) blir låst precis innan det används och släppt direkt därefter.
- 2-fas-låsning
- konservativ 2-fas-låsning

Svar:

- låsning: hjälper ingenting (samma operationsföljder är fortfarande möjliga).
- 2-fas-låsning: Förhindrar felaktig ändring; istället blir det dock **deadlock**: Efter de första 3 stegen i OF2 är C låst av T1 och B låst av T2. Nu vill T1 låsa B och T2 låsa C och de får inte släppa de objekt de redan har låsta → ömsesidigt väntande :-(
- konservativ 2-fas-låsning: (Alla dataobjekt måste låsas på en gång, i början.) Förhindrar felaktig ändring **och** deadlock.

– **4p.**

9 poäng.

Uppgift 5. Givet relation $r(A, B, C, D, E)$ med funktionella beroenden

1. $AB \rightarrow C$
2. $EB \rightarrow A$
3. $C \rightarrow E$
4. $E \rightarrow DB$

(a) Bestäm alla nycklar till relationen r .

Svar: $\{A, B\}$, $\{C\}$ och $\{E\}$.

3p.

(b) Uppfyller r Boyce-Codd-normalformen? Motivera!

Svar: Ja. Alla givna beroenden uppfyller BCNF, eftersom alla vänstersidor är övernycklar.

2p.

(c) Uppfyller r tredje normalformen? Motivera!

Svar: Ja. BCNF medför 3NF.

2p.

(d) Vi vill skapa ovanstående tabell r i SQL med:

```
create table R
( A char,
  B char,
  C char,
  D char,
  E char,
  ***
);
```

I stället för *** ska det finnas rader som garanterar alla nyckelvillkor från uppgift (a). Hur ser dessa rader ut?

Svar:

```
primary key (A,B),
unique (C),
unique (E)
```

(eller också med C eller E som primärnyckel istället).

3p.

(e) När vi nu har lagt in dessa rader från uppgift (d), är det då möjligt att mata in data som bryter mot något av de funktionella beroendena ovan?

Svar: Nej (p.g.a. att vi har BCNF).

1p.

11 poäng.

Uppgift 6. Använd Entity-Relationship-modellen för att beskriva en databas. Välj ett lämpligt exempel på en databas (t. ex. en skola, ett företag, en idrottsförening, ...).

(a) Gör en beskrivning av databasen i vanliga ord: vad den innehåller, hur innehållet inbördes hänger ihop, vilka riktighetsvillkor som ska gälla.

- (b) Översätt din beskrivning från del (a) till ER-diagram. Visa tydligt hur de enskilda påståenden i textbeskrivningen motsvaras av elementen i ER-diagrammet.

Obs! Exemplet ska vara **lagom** stort för att visa att du kan ER-modellen och dess uttrycksmöjligheter; det ska inte ta upp hela din tid på den här tentan. En textbeskrivning på ca. en halv till en sida, och omkring 3 till 6 entiteter i diagrammet blir nog lagom mycket.

Svar: Det blir full poäng om de flesta av ER-modellens uttryckselement finns med och om det framgår ur texten och användningssättet att eleven har förstått hur dessa element används i ER-modellen. Om det är tveksamt huruvida eleven "kan ER-modellen" som jag uttryckte i uppgiften, då kan även varje punkt i följande lista räknas som **1p**.

- entiteter (rektanglar)
- attribut (ovaler)
- samband (rutor)
- unika attribut = nyckelattribut (attributnamn understruket)
- sammansatta attribut (ovaler på oval)
- flervärde-attribut (dubbeloval)
- ett härlett attribut (punkterad oval)
- ett "måste"-samband (dubbelstreck)
- attribut på ett samband (oval på ruta)
- 1:1-samband (två pilhuvuden)
- 1:N-samband (ett pilhuvud; måste vara åt rätt håll!)
- M:N-samband (inget pilhuvud)
- ternärt samband (ruta mellan tre rektanglar)
- rekursivt samband (ruta mellan en rektangel och sig själv)
- svag entitet (dubbel rektangel) med partiell nyckel (punkterad understruket) och identifierande samband (dubbelruta; måste vara 1:N) till identifierande ägare. – (Ger **2p**. om allt är rätt!)

12 poäng.