Eksamen Inf3100 2012 Løsningsforslag Sensorveiledning

Vareplassering(reolnr, hyllenr, vare, antall, avdeling)

Innkjøp(vare, dato, antall, enhetspris, bestiller, avdeling)

I

(i) FDene i Vareplassering:

FD1. reolnr → avdeling
FD2. vare → avdeling
} gitt i oppgaveteksten

FD3. reolnr, hyllenr → vare
FD4. reolnr, hyllenr → antall
} fra primærnøkkelen

Kandidatnøkler: reolnr, hyllenr må være med i enhver kandidatnøkkel fordi de ikke forekommer i noen høyreside. Altså er primærnøkkelen (reolnr, hyllenr) eneste kandidatnøkkel.

FD1 bryter 2NF: reolnr er ikke en supernøkkel, avdeling er ikke et nøkkelattributt, reolnr er ekte delmengde av en kandidatnøkkel. Så INF.

FD2 bryter 3NF, men oppfyller 2NF: vare er ikke en supernøkkel, avdeling er ikke et nøkkelattributt, men vare er ikke ekte delmengde av en kandidatnøkkel.

FD3 og FD4 er på BCNF: (reolnr, hyllenr) er kandidatnøkkel (og supernøkkel).

(ii) Vareplassering(reolnr, hyllenr, vare, antall, avdeling)

Dekomponerer i hht. reolnr → avdeling.
reolnr$^+$ = reolnr, avdeling

R1(reolnr, avdeling) R2(reolnr, hyllenr, vare, antall)

Siden vare → avdeling går på tvers av R1 og R2, er dekomposisjonen ikke FD-bevarende.

Alternativt:

Vareplassering(reolnr, hyllenr, vare, antall, avdeling)

Dekomponerer i hht. vare → avdeling.
vare$^+$ = vare, avdeling

S1(vare, avdeling) S2(reolnr, hyllenr, vare, antall)

Siden reolnr → avdeling går på tvers av S1 og S2, er dekomposisjonen ikke FD-bevarende.

I

(iii)

FDer i Innkjøp:

- Fra primærnøkkelen:

  vare, dato → antall, enhetspris, bestiller, avdeling

- Hver vare bestilles av én avdeling:

  vare → avdeling

- Hver person er ansatt på én avdeling:

  bestiller → avdeling

(iv)

	A	B	C	D	E	F	G	H
ABFG	a	b	~~x_1~~c	~~x_1~~d	~~e_1~~e	f	g	~~h_1~~h
ACFG	a	~~b_2~~b	c	~~x_2~~d	~~e_2~~e	f	g	~~b_2~~ ~~h_1~~h
CDE	~~g_3~~a	~~b_3~~b	c	d	e	f_3	~~g_3~~g	~~b_3~~ ~~h_1~~h
EFH	~~g_4~~a	~~b_4~~b	~~g_4~~c	~~x_4~~d	e	f	~~g_4~~g	h

AE → D, AFG → C, B → E, C → AB, CD → H, EF → CDG

Chasealgoritmen gir tupler uten indekserte elementer (f.eks. første tuppel), følgelig er dekomposisjonen tapsfri.

II

(i) Den/de varene som tar opp mest hylleplass:

Hvor mange hyller hver vare opptar:

```
create view Hylleant as
    select vare, count(*) as hant
    from Vareplassering
    group by vare;
```

De varene som opptar flest hyller:

```
select *
from Hylleant h1
where h1.hant >= all (select hant from Hylleant);
```

(ii) Gjennomsnittlig innkjøpspris for bananer i 2011:

Oversikt over totalpriser på bananer i 2011:

```
create view Totpris as
    select antall, antall * enhetspris as totpris
    from Innkjop
    where dato > 2010 and dato < 2012
        and vare = 'bananer';
```

Svaret blir da:

```
select sum(totpris) / sum(antall) as snittpris
from Totpris;
```

II

(iii) Reolen der summen av varenes verdi er høyest:

Nyeste innkjøpspriser:

```
create view Nyestepris as
  select k1.vare, k1.enhetspris
  from Innkjøp k1
  where k1.dato = (select max(k2.dato)
                   from Innkjøp k2
                   where k2.vare = k1.vare);
```

Hyllenes verdi:

```
create view Hylleverdi as
  select v.reolnr, v.hyllenr, v.antall * p.enhetspris as verdi
  from Vareplassering v, Nyestepris p
  where v.vare = p.vare;
```

Reolenes verdi:

```
create view Reolverdi as
  select reolnr, sum(verdi) as reolverdi
  from Hylleverdi
  group by reolnr;
```

Svaret blir da:

```
select r1.reolnr, r1.reolverdi
from Reolverdi r1
where r1.reolverdi = (select max(r2.reolverdi)
                      from Reolverdi r2);
```

II

(iv) Alle bestillinger av varer i en rad er foretatt av én person:

```
select v.radnr
from Vareplassering v, Innkjøp k
where v.vare = k.vare
group by v.radnr
having count(distinct k.bestiller) = 1
```

(v) Personer som har foreslått mer enn ti innkjøp og der alle bestillingene er av forskjellige varer:

```
select bestiller
from Innkjøp
group by bestiller
having count(*) > 10 and count(vare) = count(distinct vare);
```

III

(i) Avdelingen i Vareplassering stemmer overens med avdelingen i Innkjøp:

$$\Pi_{\text{Vareplassering.vare, Vareplassering.avdeling, Innkjøp.avdeling}} \left(\sigma_{\text{Vareplassering.avdeling} <> \text{Innkjøp.avdeling}} \left(\text{Vareplassering} \bowtie_{\text{Vareplassering.vare} = \text{Innkjøp.vare}} \text{Innkjøp} \right) \right)$$

Det står ikke eksplisitt hva vi skal gi som tilbakemelding der vi finner avvik, men her har vi valgt ut varens navn og de to avdelingene som er angitt på varen. Her er det altså opp til den enkelte om man bare returnerer $\sigma(V \bowtie I)$ "=" eller noe annet.

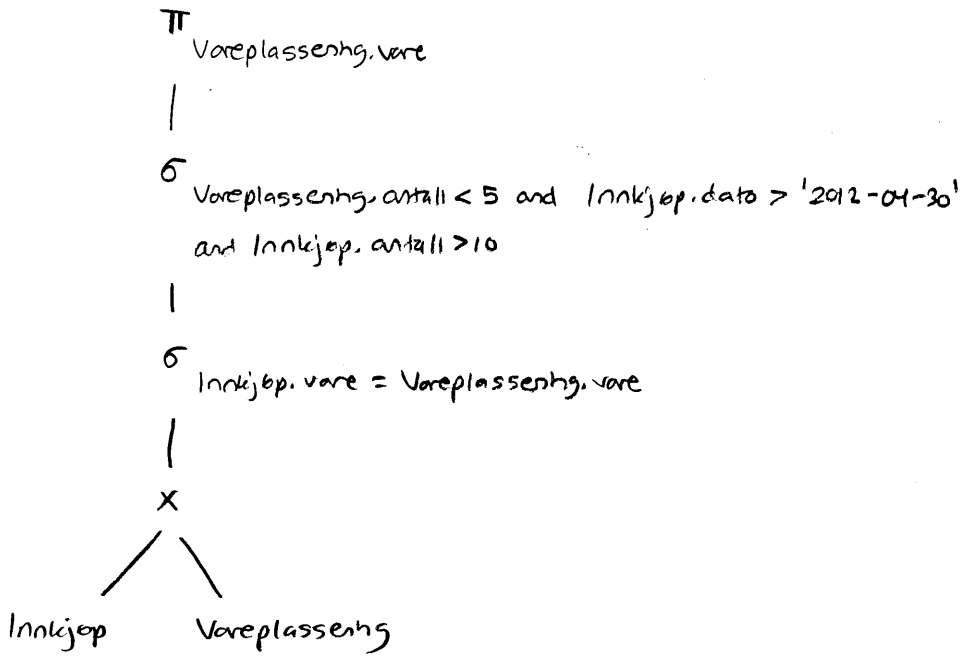
Finner de hvor det er avvik

NB! Naturlig join kan ikke brukes, ferdi da joines det på (vare, antall, avdeling) – og det blir helt feil. Det skal bare joines på 'vare'.

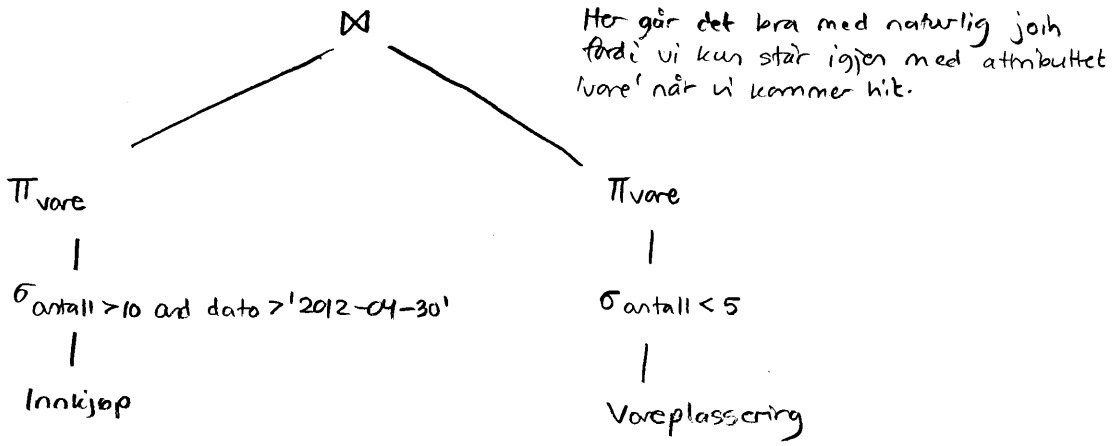
III

(ii)

Opprinnelig uttrykk:



Optimert: Skyver π og σ nedover, men som i III (i) er det viktig at man ikke har en naturlig join med mindre felles attributt bare er 'vare'. Ellers må en equijoin eller en thetajoin benyttes (dvs. en join hvor man angir eksplisitt joinkriteriet).



IV

Her er det ganske mye å lese, men relativt raskt å gjøre oppgavene hvis man bare skjønner dette. Det er slik at stoffet i oppgaven er fra det i læreboken som er merket med 'statarisk', men de har ikke nødvendigvis særlig trening med akkurat denne varianten av låser, derfor valgte jeg å skrive ganske eksplisitt hvordan låsene virker og hvordan de skal tenke. Men spørsmålet er selvfølgelig hva de måtte å forelegge av dette på den tiden de har til rådighet. Resten av oppgavesettet er imidlertid ganske rett fram og "mainstream".

(i) Varsellås på relasjonsnivå, vanlig lås på tuplene:

Type 1: IX(Flyavgang); X(a); r(a);
IX(Bestilling); X(b); w(b); w(a);
U(a); U(Flyavgang); U(b); U(Bestilling);

Oppløsing i en eller annen rekkefølge.
U(b) og U(Bestilling) kan komme allerede rett etter w(b).

Type 2: IX(Bestilling); X(b); r(b); w(b);
IX(Flyavgang); X(a); r(a); w(a);
U(b); U(Bestilling); U(a); U(Flyavgang);

Oppløsing i en eller annen rekkefølge.
U(b) og U(Bestilling) kan komme allerede rett etter X(a).

Også andre rekkefølger er mulige sålenge 2PL følges, dvs. man må ta alle nødvendige låser før første oppløsing skjer. Dessuten må man ta en IX-lås på den relasjonen et tuppel befinner seg i før man tar en X-lås på selve tuppelet.

IV

(ii) Det som ~~kan skape~~ kunne ha skapt en vranglåssituasjon, ~~og~~ var hvis person 1 og person 2 ~~prøver~~ prøvde å aksessere samme tuppel i Flyavgang samtidig. (De vil ha separate tupler i Bestilling, så disse utgjør ikke noe problem.) Imidlertid vil ingen av de to hindres fra å ta IX-lås på Flyavgang siden kompatibilitetsmatrisen tillater flere å ta skriveversellås samtidig. Men da er det slik at den av de to som ~~får~~ X-låst på det aktuelle tuppelet i Flyavgang først, også alltid vil kunne fullføre sin transaksjon. Så det blir aldri noen vranglås av ~~dette~~. (Vranglås kan bare oppstå hvis to eller flere transaksjoner venter gjensidig på hverandre, men da må det også være to eller flere eksklusive ressurser/elementer som de kjemper om samtidig, men her er det bare én ressurs.)