

Innlevering 2 INF3100

Endre Wullum
endrewu@ulrik.uio.no

8. mai 2015

Oppgave 1

(i)

Boligsalg(salgsnr, mnr, adr, bolignr, salgsdato, boligtype, areal, pris)

FDer:

$mnr \rightarrow adr, bolignr$

$adr, bolignr \rightarrow mnr$

$salgsnr \rightarrow mnr, adr, bolignr, salgsdato, boligtype, areal, pris$

(ii)

$mnr \rightarrow adr, bolignr$ er 2NF fordi adr, bolignr ikke er en supernøkkel, ikke er nøkkelattributter men ikke er del av noen kandidatnøkkel

$adr, bolignr \rightarrow mnr$ er 2NF fordi adr, bolignr ikke er en supernøkkel, ikke er nøkkelattributter men ikke er del av noen kandidatnøkkel

$salgsnr \rightarrow mnr, adr, bolignr, salgsdato, boligtype, areal, pris$ er på BCNF fordi salgsnr er supernøkkel

(iii)

Adresse(mnr, adr, bolignr)

Boligsalg(salgsnr, mnr, salgsdato, boligtype, areal, pris)

Oppgave 2

(i)

```
create view Kjoper as  
select navn, personnr, mnr, adr, bolignr  
from Person natural join Salgspart natural join Boligsalg  
where salgsrolle = 'kjoper';
```

```
create view Selger as  
select navn, personnr, mnr, adr, bolignr  
from Person natural join Salgspart natural join Boligsalg  
where salgsrolle = 'selger';
```

```
create view Megler as  
select navn, personnr, mnr, adr, bolignr  
from Person natural join Salgspart natural join Boligsalg  
where salgsrolle = 'megler';
```

```
select navn, personnr, adr, bolignr  
from Kjoper k, Selger s, Megler m  
where k.mnr = s.mnr and s.mnr = m.mnr  
      and k.personnr = s.personnr and s.personnr = m.personnr;
```

(ii)

```
select mnr, count(*)  
from Boligtype b1 join Boligtype b2 on b2.salgsnr = (  
    select min(salgsnr)  
    from Boligsalg  
    where salgsnr > b1.salgsnr)  
where b1.salgsnr <> b2.salgsnr;
```

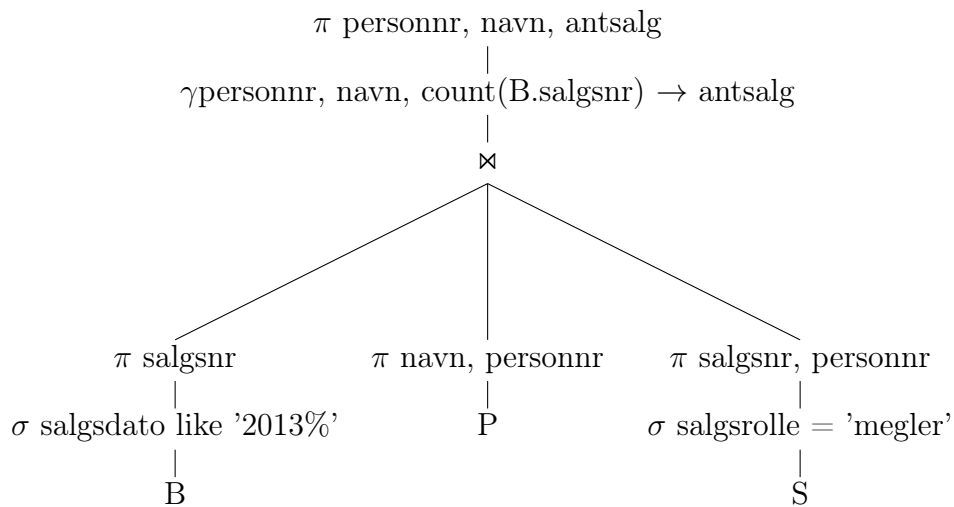
(iii)

```
select distinct mnr  
from Boligsalg  
where mnr not in (  
    select mnr  
    from Boligsalg natural join Salgspart natural join Person  
    where salgsrolle = 'megler');
```

Selv om, rent logisk må man spørre seg om en bolig solgt uten bruk av en megler ville havnet i en boligsalgsdatabase kurert av boligmeglere.

Oppgave 3

(i)



(ii)

$$\sigma_{\text{antall} > 1}(\gamma_{\text{salgsnr}, \text{count}(\text{salgsrolle} \rightarrow \text{antall}(\sigma_{\text{salgsrolle} = \text{'megler'}})^{\text{salgspart}})) = \emptyset$$

4

(i)

$$T1 = r1(a); r1(b); l1(a); w1(a); l1(b); w1(b); u1(a, b)$$

$$T2 = r2(c); r2(a); l2(c); w2(c); l2(a); w2(a); u2(a, c)$$

$$T3 = r3(a); r3(b); r3(c); l3(c); w3(c); u3(c)$$

(ii)

T_1	T_2	T_3
$r_1(a)$ $r_1(b)$	$r_2(c)$ $r_2(a)$	$r_3(a)$ $r_3(b)$ $r_3(c)$
$l_1(a)$ $w_1(a)$ $l_1(b)$ $w_1(b)$ c_1 $u_1(a, b)$	$l_2(c)$ $w_2(c)$ $l_2(a) - \text{avslått}$ a_2 $u_2(c)$	$l_3(c)$ $w_3(c)$ c_3 $u_3(c)$

$l_2(a)$ blir avslått fordi T_1 allerede har skrevet til a .

5

(i)

2 diskkræsje, d_i og d_{i+1} hvor i beregnes modulo $2m+1$, er tilstrekkelig til å gi varig tap av data, gitt at de to diskene som kræsjer, mellom seg, inneholder begge speilene av samme stripe hvor k er en tilfeldig valgt disk i RAIDet.

(ii)

Det høyeste antallet diskkræsjer som ikke medfører varig tap av data er m . Forutsatt at disse diskkræsjeene ikke påvirker to disk, d_i og d_{i+1} som speiler samme stripe.

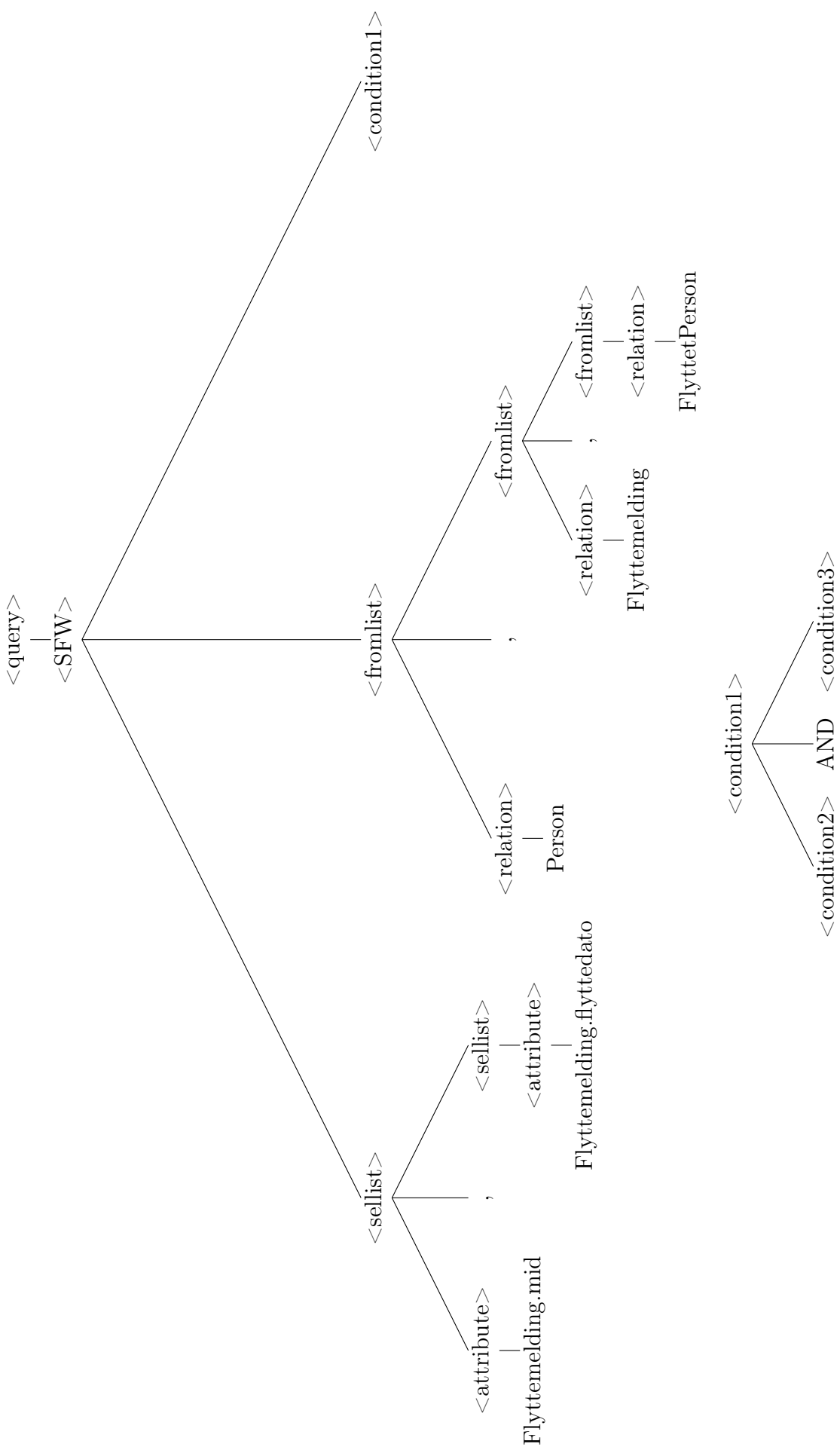
Oppgave 6

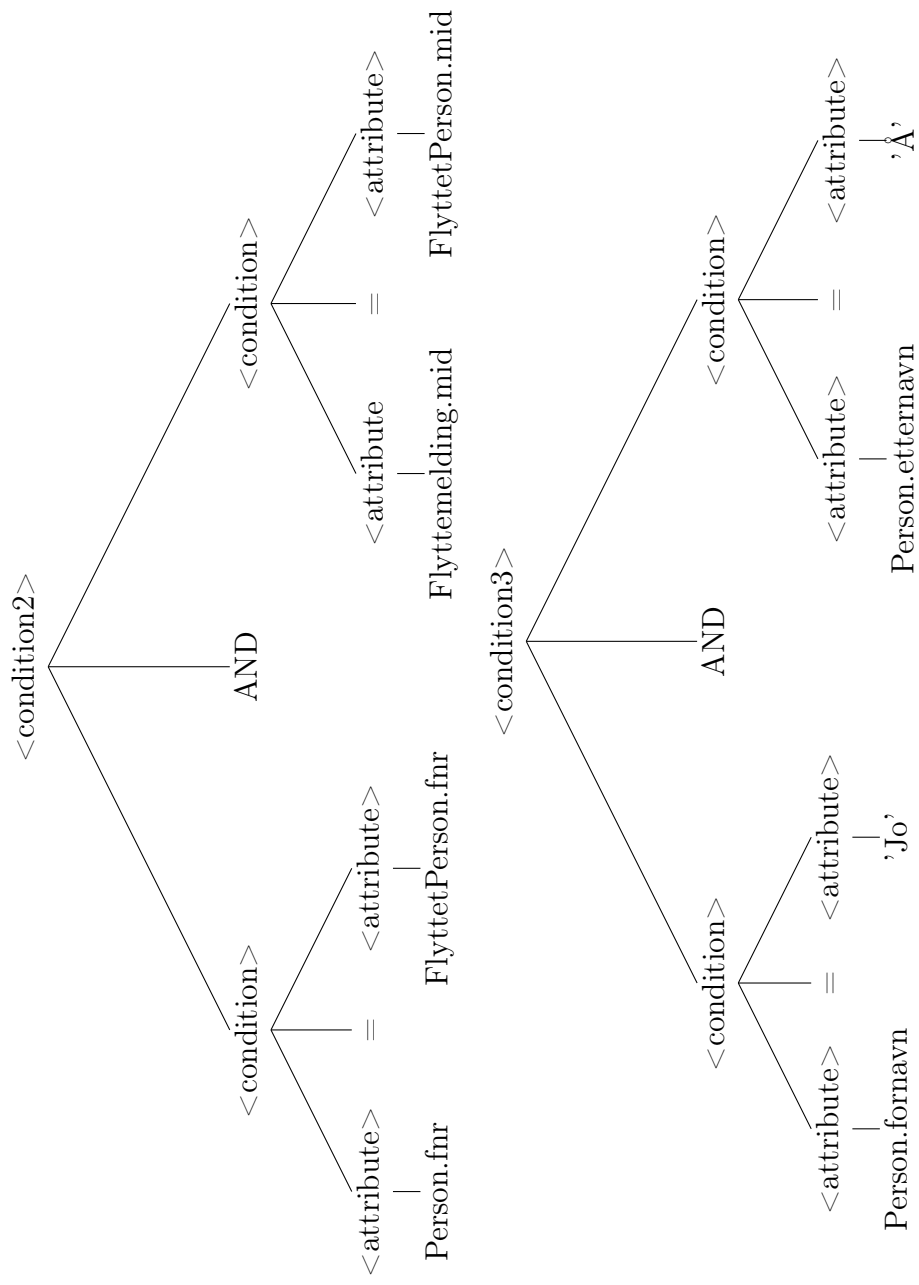
a

LaTeX-pakken jeg bruker for å tegne trær (qtree) har en begrensning hvor en node kun kan ha 5 barn. Derfor har jeg valgt å kutte nodene SELECT, FROM og WHERE fra <SFW>. I dette treet anser jeg disse som implisitte at de kommer barnene til <SFW> kommer i rekkefølgen SELECT <sellist> FROM <fromlist> WHERE <condition>.

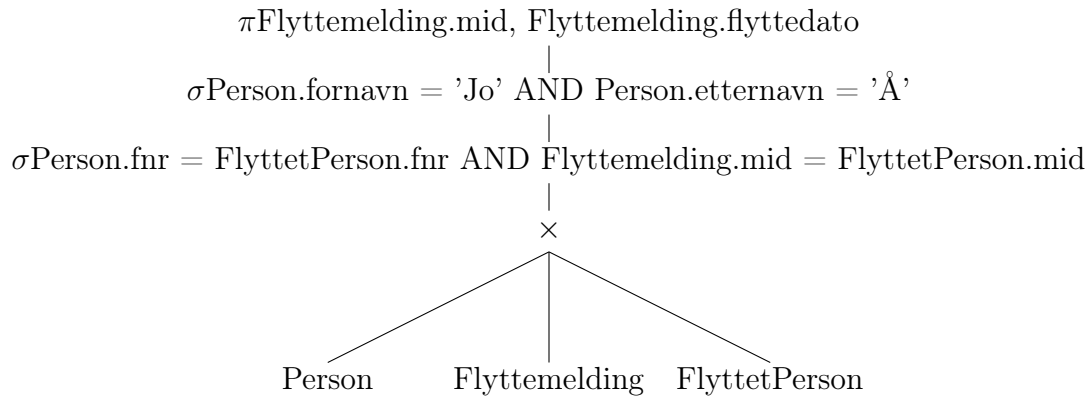
Videre, grunnet plassbegrensning har jeg måttet skille treet i flere deler i <condition>-delen av treet. Jeg har prøvd å gjøre dette så oversiktlig som mulig, så treet som begynner på rotnoden <condition1> hører til hvor <condition1> er en løvnode osv.

Jeg har også balansert <condition>-subtreet, og er ikke helt sikker på om det er lov eller anbefalt. Håper det er mulig å få en avklaring på det.

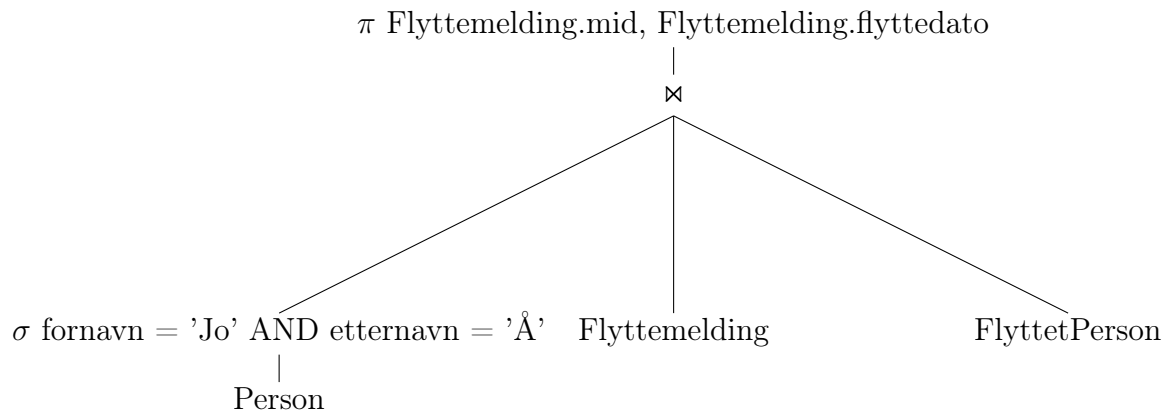




b

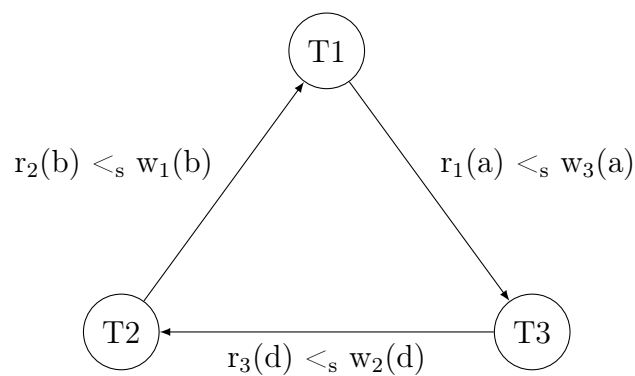


c



Oppgave 7

a



Siden presedensgrafen blir sirkulær er ikke S_1 konfliktserialiserbar.

b

$T1 = sl1(a); r1(a); sl1(b); r1(b); xl1(b); w1(b); sl1(c); r1(c); xl1(c); w1(c); u1(a, b, c)$

$T2 = sl2(b); r2(b); sl2(d); r2(d); xl2(d); w2(d); u2(b, d)$

$T3 = sl3(d); r3(d); sl3(a); r3(a); xl3(a); w3(a); u3(d, a)$

c

T_1	T_2	T_3
$sl_1(a)$ $r_1(a)$		
		$sl_3(d)$ $r_3(d)$
$sl_1(b)$ $r_1(b)$		
	$sl_2(b)$ $r_2(b)$	
$xl_1(b)$ - avslått		$sl_3(a)$ $r_3(a)$ $xl_3(a)$ - avslått
	$sl_2(d)$ $r_2(d)$ $xl_2(d)$ - avslått	

Og vi ender opp med en vranglås hvor $T1$ venter på $T2$, $T2$ venter på $T3$ og $T3$ venter på $T1$.

Oppgave 8

a

Siden $T1$ er eldre enn $T2$ får den vente. Siden $T2$ er eldre enn $T3$ får den vente. Fordi $T3$ er yngre enn $T1$ må den rulle tilbake så de ventende transaksjonene får fullføre. Dette skjer i steg 9.

b

Fordi $T1$ er eldre enn $T2$ vil $T1$ skade $T2$ og $T2$ må rulle tilbake. Dette skjer i steg 7. $T3$ ender opp med å vente på $T1$.

Oppgave 9

a

<T1,start>
<T1,b,17>
<T1,c,19>
<T1,commit>

b

Alle endringer skal skrives til logg før endringen utføres på disk.
Alle endringer skal være utført på disk før commit skrives til logg.