

Oppg. 1

1a

(i) $\left. \begin{array}{l} \text{fødselsNr} \rightarrow \text{søknadsNr, navn, karaktersnitt} \\ \text{søknadsNr} \rightarrow \text{fødselsNr} \end{array} \right\} \text{ i Elev}$

$\left. \begin{array}{l} \text{søknadsNr} \Rightarrow \text{skole} \\ (\text{søknadsNr} \Rightarrow \text{linje} - \text{uødvendig, for} \\ \text{følger av søknadsNr} \Rightarrow \text{skole}) \end{array} \right\} \text{ i Søknad}$

(ii) Kandidatnøkler i Elev: Siden

$\text{fødselsNr}^{\dagger} = \text{fødselsNr, søknadsNr, navn, karaktersnitt}$
 $\text{søknadsNr}^{\dagger} = \text{fødselsNr}$

er hver av $\{\text{fødselsNr}\}$ og $\{\text{søknadsNr}\}$ kandidatnøkler.
 (Dette er de eneste; de øvrige attributtene forekommer bare i høyresider)

Da begge venstresidene i de to FDene i Elev er kandidatnøkler, er de også supernøkler, og oppfyller følgende kravene til BCNF.
 (Det Ans'nges ikke trivielle MVDer i Elev, så 4NF er også oppfylt.)

(iii) Kandidatnøkler i Søknad:

Det er ingen ikke trivielle FDer i Søknad, så alle attributtene $\{\text{søknadsNr, skole, linje}\}$ danner en kandidatnøkkel (den eneste).

MVDen er ikke triviell (høyresiden ikke del av venstresiden, vs + hs ikke alle attr.)

Venstresiden i $\text{søknadsNr} \Rightarrow \text{skole}$ er ikke en kandidatnøkkel og altså ikke en supernøkkel, så det er brudd på 4NF. Derfor er BCNF oppfylt.

1b

```

(i) select s, skole, L, linje, count(s, søknadsNr) as antallsøkere
    from Skolesøknad s, Linjesøknad l
    where s.søknadsNr = L.søknadsNr
    group by s.skole, L.linje
    order by antallsøkere desc;

```

```

(ii) select
    from Tilbud t
    where (t.skole, t.linje) not in (select s.skole, L.linje
                                     from Skolesøknad s, Linjesøknad l
                                     where s.søknadsNr = L.søknadsNr);

```

ELLER F. EKST.

→ De skole-linje-kombinasjonene som det fins søknader til

```

(select skole, linje
 from Tilbud)
except all
(select skole, linje from (Skolesøknad natural join Linjesøknad));

```

→ Kan bruke except all fordi første select er en mengde siden (skole, linje) er primærnøkkel.

1b

(iii)

```

select
from Elev e
where not exists (
  (select s.skole, l.linje
   from Skolesøknad s, Linjesøknad l, Søknad a
   where a.fødselsNr = e.fødselsNr and
         a.søknadsNr = s.søknadsNr and
         a.søknadsNr = l.søknadsNr)

```

```

intersect
(select skole, linje
 from Tilbud);

```

Skulle gå bra med intersect all også, for select skole, linje returnerer en mengde

De skole-linje-kombinasjonene som tilbys

De skole-linje-kombinasjonene som e har søkt

1c

```

τ ontallsøkere desc (γ skole, linje, count(søknadsNr) → ontallsøkere
                    (Skolesøknad ⋈ Linjesøknad))

```

↑
 Kan eventuelt ha med τ skole, linje, ontallsøkere her, men det er unødvendig fordi det er disse som uansett leveres i svaret fra γ.

Oppgave 2

- (i) En tapsfri dekomposisjon er en oppsplitting i relasjoner der oppsplittingen er slik at det ikke dannes falske tupler ved naturlig join.

Litt mer presist: En tapsfri dekomposisjon av en relasjon R er en samling relasjoner R_1, \dots, R_n hvor vi for enhver lovlig instans av R har at hvis instansen projiseres på (attributtene i) R_1, \dots, R_n og vi deretter foretar en naturlig join av projeksjonene igjen, så gjenstapes den opprinnelige instansen nøyaktig.

(ii)

	A	B	C	D	E	F	G	H
ABCE	a	b	c	$\not\exists_1 d$	e	$\not\exists_1 f$	$\exists_1 g$	$\not\exists_1 h$
ACFGH	a	b_2	c	$\not\exists_2 d$	$\not\exists_2 e$	f	g	h
BCDFG	$\not\exists_3 a$	b	c	d	$\not\exists_3 \not\exists_2 e$	f	g	$\not\exists_3 h$
BDEG	$\not\exists_4 a$	b	c_4	d	e	$\not\exists_4 \not\exists_1 f$	g	h_4

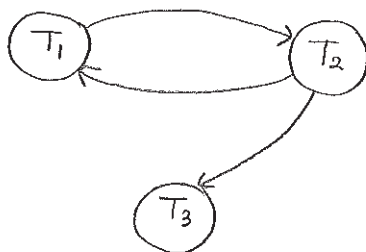
Braker Chasealgoritmen med de oppgitte FDene. Siden minst én rad da får kun bokstaver uten indekser, er dekomposisjonen tapsfri.

(Det er ikke nødvendig å fortsatte chasen etter at en slik rad har oppstått; f.eks. kunne vi ha stanset på et tidspunkt der a_4 fortsatt var en verdi, men hver rad i alt var uten indekser.)

Oppgave 3

3a

(i)



Siden presedensgraften har en sykkel, er S ikke konfliktserialiserbar.

$(T_1 \rightarrow T_2$ fordi f. eks. $w_1(b) <_S r_2(b)$)

$T_2 \rightarrow T_1$ \neg $w_2(c) <_S r_1(c)$

$T_2 \rightarrow T_3$ \neg $w_2(c) <_S r_3(c)$)

(ii)

	T_1	T_2	T_3
$L_1(a)$			
$r_1(a)$			
$L_1(b)$			
$r_1(b)$			
$w_1(b)$			
		$L_2(c)$	
		$r_2(c)$	
		$L_2(b)$ - vent på T_1	
			$L_3(d)$
			$r_3(d)$

$T_1: L_1(a); r_1(a); L_1(b); r_1(b); w_1(b); L_1(c); u_1(b); r_1(c); w_1(c); u_1(c); w_1(a); u_1(a)$
 $T_2: L_2(c); r_2(c); L_2(b); r_2(b); w_2(c); u_2(c); w_2(b); u_2(b)$
 $T_3: L_3(d); r_3(d); L_3(c); r_3(c); w_3(c); u_3(c); w_3(d); u_3(d)$

$L_1(c)$ -
vent på T_2 ;
vranglås
mot T_2

$L_3(c)$ -
vent på T_2 ;
 T_3 blir også
rammet av
vranglåsen
mellom
 T_1 og T_2

III b

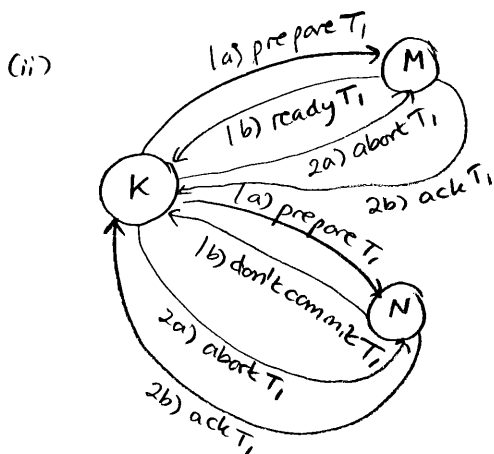
- (i) 2PC brukes for å gjennomføre commit for en transaksjon som er distribuert over flere noder. En av nodene er koordinatør og bestemmer når 2PC-protokollen skal påbegynnes (typisk når koordinatør har eksekvert sin del av den distribuerte transaksjonen og vet om den kan committe eller må abortere).

Fase 1: Koordinatør spør samtlige deltakere om de er klare til å committe. Den enkelte node avslutter sin del av transaksjonen og tar en selvstendig beslutning om commit/abort. Dersom den kan committe, sender den en melding om dette og venter på fase 2. Hvis ikke, aborterer den og sender en melding om dette. (Deretter er den klar til å behandle meldinger for fase 2.)

Koordinatør samler inn svar fra samtlige (eller venter til en gitt tid er gått).

Fase 2: Hvis samtlige svarte at de kan committe, sender koordinatør en melding til alle om at de skal gjøre dette. Hvis minst én svarte abort, eller hvis ikke alle svarte på fase 1 i tide, sender koordinatør melding til alle om å abortere.

(I tillegg er det en hel del som kan sies om hva som må gjøres hvis én eller flere noder går ned mens protokollen pågår.)



(Hvilke navn som brukes på meldingene, er ikke så viktig så lenge semantikken fremgår.)

(I tillegg skrevet K, M, N noen loggposter blealt, disse vises ikke her.)

- (iii) M sender en melding til K hvor den ber om å få vite hva K sendte i fase 2. Deretter avslutter den i henhold til abort-meldingen. (Flere mulige svar her: M kan f.eks. spørre N.)