

Person(fnr, fornavn, mellomnavn, etternavn)

Flyttemelding(mid, fnr, flyttedato, fraadr, tiladr)

(A) mid  $\rightarrow$  flyttedato, fraadr, tiladr

(B) fnr, flyttedato  $\rightarrow$  mid

I

(i) Kandidatmøker i Flyttemelding:

fnr forekommer ikke på noen høyreside og må derfor være med i alle knk.

fnr, flyttedato  $\rightarrow$  fnr, flyttedato, mid, fraadr, tiladr

fnr, mid  $\rightarrow$  fnr, mid, flyttedato, fraadr, tiladr

Det er følgende to kandidatmøker: (fnr, flyttedato) og (fnr, mid).

(ii) Normalformer:

mid  $\rightarrow$  flyttedato

mid er ikke supermøkkel, så bryter BCNF  
flyttedato er nøkkelattributt, så oppfyller 3NF

mid  $\rightarrow$  fraadr  
mid  $\rightarrow$  tiladr

mid er ikke supermøkkel, så bryter BCNF  
fraadr/tiladr er ikke nøkkelattributt, så bryter 3NF

mid er del av en kandidatmøkkel, så bryter 2NF

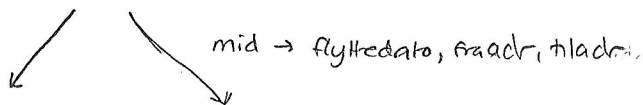
fnr, flyttedato  $\rightarrow$  mid

(fnr, flyttedato) er supermøkkel, så oppfyller BCNF

Totalt brytes 2NF, så Flyttemelding er på 1NF, men ikke 2NF.

(iii) Tar utgangspunkt i de FDer under (ii) som bryter BCNF. Samler dem først til en igjen (som i (A)). Dekomposisjonen til BCNF går da etter slik:

Flyttemelding(mid, fnr, flyttedato, fraadr, tiladr)



A1(mid, flyttedato, fraadr, tiladr) A2(mid, fnr)

FDer:

mid  $\rightarrow$  flyttedato, fraadr, tiladr  
(BCNF i A1)

FDer:

(Ingen ikke-triviale)

I tillegg gjelder fortsatt fnr, flyttedato  $\rightarrow$  mid som går på tross av A1 og A2, så dekomposisjonen er ikke FD-bevarende.

I

(iv) Bruker chasealgoritmen:

	mid	fnr	flyttdato	fnradr	tladr
61	M	f <sub>1</sub>	d <sub>1</sub> D	FA	TA
62	M	F	D	f <sub>2</sub> FA	t <sub>2</sub> TA

Siden andre rad er uten inkluderte variable, er dekomposisjonen tapsfri.

(v) (61,62) er FD-bevarende siden ingen av FDene går på tvers. Samtidig bryter 62 BCNF (den er på EKNF):

(61,62) er altså den høyeste normalformen vi kan få uten å måtte sjekke FDe på tvers av relasjoner, så den representerer minst mulig overlappning når FD-bevning. Fordelen fremfor dekomposisjonen i (iii) er at FDene kan sjekkes intern i relasjonene; for å sjekke dette i (iii), må vi først joine A1 og A2. Ulempen er at det er noe dobbeltlagning i 62.

Sannsynligvis vil man foretrekke å ha relasjonene på formen (61,62) (enkle å sjekke) - eller tilogmed den opprinnelige (Flyttemelding) for å slippe joinoperasjoner hver gang relasjonene brukes.

## II

(i) create table Flyttemelding (  
 mid integer,  
 fnr integer references Person(fnr),  
 flyttdato date not null,  
 fraadr varchar(100),  
 tiladr varchar(100) not null,  
primary key (mid, fnr),  
unique (fnr, flyttdato),  
check (fraadr <> tiladr)  
 );

(ii) select distinct f1.mid  
from Flyttemelding f1, Flyttemelding f2  
where f1.mid = f2.mid and -  
 (f1.flyttdato <> f2.flyttdato or  
 f1.fraadr <> f2.fraadr or  
 f1.tiladr <> f2.tiladr);

} sammenligner to med  
 samme mid, de  
 skal ha like flyttdato,  
 fraadr og tiladr. Hvis  
 ikke, skriv ut mid.

(iii) select f1.fnr, f1.tiladr  
from Flyttemelding f1, Person p  
where f1.fnr = p.fnr and  
 p.foemaun = 'Marius Sophus' and  
 p.etternam = 'Lie' and  
 f1.flyttdato = (select max(f2.flyttdato)  
from Flyttemelding f2  
where f2.fnr = f1.fnr);  
order by f1.fnr;

} alle flyttemeldinger  
 for fnr;  
 finn siste flyttemelding  
 basert på dato

(iv) select f1.fnr, f1.mid, f2.mid  
from Flyttemelding f1, Flyttemelding f2  
where f1.fnr = f2.fnr and f1.flyttdato < f2.flyttdato and  
not exists (select f3.mid  
from Flyttdato f3  
where f3.fnr = f1.fnr and  
 f3.flyttdato > f1.flyttdato and  
 f3.flyttdato < f2.flyttdato)  
and f1.tiladr <> f2.fraadr;

} f1 er før f2

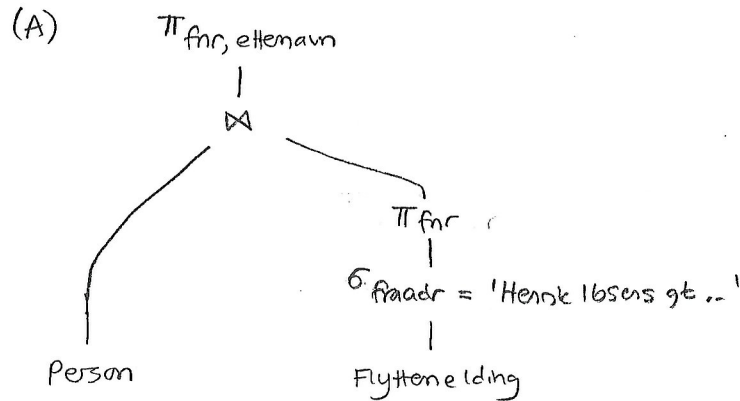
} det er ingen  
 flyttemeldinger  
 mellom disse  
 (for fnr)

} de to adressene  
 stemmer ikke  
 overens

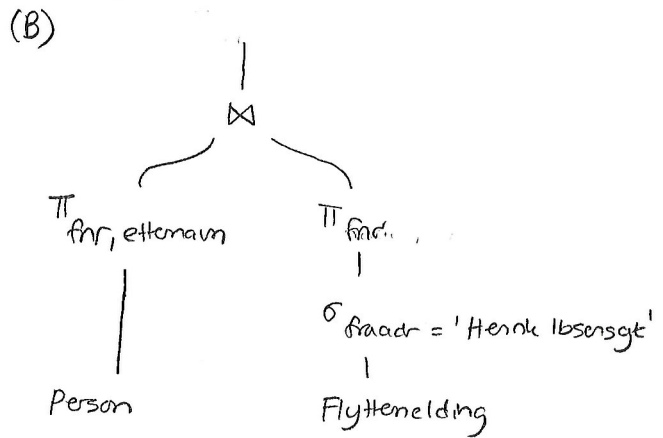
III

(i) Uttrykket finner fødselsnummer og etternavn til alle personer som har flyttet fra Henrik Ibsens gate 26.

(ii) Kan skyve seleksjon nedover, og introdusere projeksjon tidlig før attributter som ikke trengs.



(En annen variant er som følger, men se iii hvorfor den kanskje ikke er like ideell:



(iii) (A) I Flyttemelding får vi ikke brukt indeksen fordi det selekteres på et attributt som ikke er indeksert.

I den naturlige joinen, derimot, kan vi utnytte indeksen på fnr i Person i joinalgoritmen, til å plukke effektivt ut bare de forekomstene som er aktuelle.

(B) I denne versjonen vil man ikke kunne nyttiggjøre seg noen av indeksene fordi projeksjonen på fnr, etternavn for Person gjør at vi deretter ikke har noen indeks på resultatsettet.

(Opprinnelig uttrykkestre: Her kan man heller ikke nyttiggjøre seg indeksene. Det kartesiske produktet kobler alle tupler i Person mot alle tupler i Flyttemelding til en stor relasjon, uten indekser.)

#### IV

- (i) I Snapshot Isolation baserer hver transaksjon seg på de verdiene som er committet på det tidspunktet transaksjonen påbegynnes (derav ordet 'snapshot'). Hvis to samtidige transaksjoner har overlappende skrivemengder, må en av dem rullles tilbake.
- (ii) Det fins tilfeller hvor SI-protokollen slipper igjennom transaksjoner, men hvor resultatet ikke kan forklares som om transaksjonene ble utført én for én. Det betyr strengt tatt at vi risikører at noen transaksjoner kan "ødelegge" for hverandre slik at databasen blir inkonsistent. Både de fenomenene som forårsaker dette, er svært sjeldne i praksis, og SI-protokollen er billig å håndtere og tillater høy samtidighet, er likevel SI-protokollen anbefalt brukt i mange (de fleste) situasjoner. (Det avhenger likevel av hvor kritisk det er hvis feil/anomalier opptrer, om SI-protokollen egner seg eller ikke.)

