

# Innlevering 1 INF3100

Endre Wullum  
endrewu@ulrik.uio.no

4. mai 2015

\*Obligen er løst i samarbeid med geirmuns og torsiguh.

## Oppgave 1

### A

Tog er på BCNF fordi for togNr er en supernøkkel og det ikke finnes flere FDer i relasjonen.

TogTabell er også på BCNF fordi den kun har en FD som er supernøkkel.

Plass er på 1NF fordi FDen  $togNr, vognNr, plassNr \rightarrow vindu$  ikke er slik at  $X$  er en supernøkkel, vindu er ikke et nøkkelattributt i relasjonen og  $X \subset K$ . Altså oppfyller FDen ingen av kravene for 2NF.

### B

	dato	togNr	vognNr	plassNr	vindu	ledig
Sete	dato <sub>1</sub>	togNr	vognNr	plassNr	vindu	ledig <sub>1</sub>
Plass	dato	togNr	vognNr	plassNr	vindu <sub>2</sub>	ledig

Siden det finnes en rad uten subscript kan vi se at dekomposisjonen er tapsfri.

De nye relasjonene er begge på BCNF fordi de oppfyller kravet om at FDen er en supernøkkel.

Et åpenbart problem med dekomposisjonen er at det er mye overlapp i de to nye relasjonene, dette vil føre til veldig mye dobbeltlagring av data, noe som ikke er ideelt.

### C

$\Pi$  vognNr (togNr = 401 and dato = 10.6.2008)<sup>Plass</sup> —

$\Pi$  vognNr (togNr = 401 and vindu = true and dato = 10.6.2008 and ledig = true)<sup>Plass  $\bowtie$  Sete</sup>

## D

La  $T = \text{TogTabell}$

$$\sigma\theta(\rho_{T_1}(T) \times \rho_{T_2}(T)) = \emptyset$$

Hvor  $\theta = (T_1.\text{togNr} = T_2.\text{togNr} \text{ and } T_1.\text{stasjon} = T_2.\text{stasjon} \text{ and } T_1.\text{ankomstTid} \neq T_2.\text{ankomstTid})$

## Oppgave 2

### A

BCF, ACF og CDEF er kandidatnøkler

### B

$\mathcal{R}$  tilfredsstillter høyest 1NF

### C

$$\mathcal{D} = \{ABF, ACF, BCDE, DG\}$$

	A	B	C	D	E	F	G
ABF	a	b	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>	f	g <sub>1</sub>
ACF	a	b <sub>2</sub>	c	d <sub>2</sub>	e <sub>2</sub>	f	g <sub>2</sub>
BCDE	a <sub>3</sub>	b	c	d	e	f <sub>3</sub>	g <sub>3</sub>
DG	a <sub>4</sub>	b <sub>4</sub>	c <sub>4</sub>	d	e <sub>4</sub>	f <sub>4</sub>	g

Det finnes ingen rader som kan bearbejdes bedre ut i fra FDene og ingen rader har kun konstanter, altså er ikke dekomposisjonen tapsfri.

## D

Dekomponering av  $\mathcal{R}(ABCDEFGG)$ :

Med utgangspunkt i  $CDE \rightarrow B$ :

$$\mathcal{R}_1(ABCDEG)$$

$$\mathcal{R}_2(CDEF)$$

Videre dekomponering av  $\mathcal{R}_1$ :

$$\mathcal{R}_3(DG)$$

$$\mathcal{R}_4(ABCDE)$$

Videre dekomponering av  $\mathcal{R}_4$ :

$\mathcal{R}_5(BA)$

$\mathcal{R}_6(BCDE)$

Nå utgjør  $\mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_5, \mathcal{R}_6$  en tapsfri dekomposisjon av  $\mathcal{R}$  på BCNF.

Dekomposisjonen er ikke FD-bevarende.

## E

$CDF \rightarrow B?$

$CDF^+ = CDFG$

Tillukningen av  $CDF$  inkluderer ikke B, altså holder ikke  $CDF \rightarrow B$

## F

A	B	C	D	E	F	G
a	b	c	d	e	f	g
a <sub>2</sub>	b <sub>2</sub>	c	d	e <sub>2</sub>	f	g <sub>2</sub>
a <sub>2</sub>	b	c	d	e	f	g
a	b <sub>2</sub>	c	d	e <sub>2</sub>	f	g

Siden de to opprinnelige radene nå er helt uten subskript så er det vist at  $CDF \rightarrow B$  holder.

## Oppgave 3

### A

```
select fornavn, etternavn, adresse
from Person p
where p.fnr in (
    select n.fnr
    from ekteskap e, navneskifte n
    where etternavn1 <> etternavn2
    and e.dato = n.dato
    and (n.fnr = e.fnr or n.fnr = e.fnr))
and e.dato >= 2000-1-1
and e.dato < 2011-1-1;
```

### B

```

select etternavn, fornavn, adresse
from person
where fnr in
    (select n.fnr
     from ekteskap e, navneskifte n
     where e.etternavn1 <> e.etternavn2
     and e.dato = n.dato
     and (
        (n.fnr = e.fnr1 and n.etternavn = e.etternavn2) or
        (n.fnr = e.fnr2 and n.etternavn = e.etternavn1)));

```

## Oppgave 4

### A

```

select maintitle, firstprodyear, count(episodeid) as episodes
from series s left join episode e on s.seriesid = e.seriesid
where firstprodyear = (select max(firstprodyear) from series) group by maintitle, first-
prodyear;

```

maintitle	firstprodyear	episodes
Tonight Show with Conan O'Brien, The	2009	0
Pacific War, The	2009	0
Untitled Star Wars TV Series	2009	0
Pacific, The	2009	0
Saka no ue no kumo	2009	0
Last Horseman, The	2009	0

\*Jeg forsto det slik at dette var utskriften oppgaven spurte etter, selv om jeg synes det virker litt uryddig å måtte bruke left join/outer join på denne måten

### B

```

select parttype, (count(personid)*1000/(select count(*) from filmparticipation)) * 0.1 as
percent
from filmparticipation
group by parttype;

```

parttype	percent
writer	7.7
costume designer	0.9
director	5.9
editor	2.9
cast	72.0
composer	2.5
producer	7.8

## C

```
select firstname, lastname
from filmparticipation fA, filmparticipation fB,
     filmparticipation fC, filmparticipation fD, person p
where fA.filmid = fB.filmid
and fC.filmid = fD.filmid
and fA.filmid <> fC.filmid
and fA.personid = fC.personid
and fB.personid = fD.personid
and fA.personid <> fB.personid
and fA.parttype = fD.parttype
and fB.parttype = fC.parttype
and fA.parttype <> fB.parttype
and p.personid = fA.personid
group by firstname, lastname
having count(distinct fB.personid) > 49;
```

firstname	lastname
Dick	Wolf
Donald P.	Bellisario
Glen A.	Larson
James L.	Brooks
Lloyd	Kaufman
Matt	Groening

\*Kjøretid for spørringen mot hele filmdatabasen endte på ca 30 minutter.

## D

Denne oppgaven har jeg ikke funnet en endelig løsning på, har hatt problemer med å finne en spørring som fungerer. I alle fall, dette er hvordan jeg tenkte at dette skulle funke:

```
select firstname, lastname
from person p,
     (select p.personid
      from filmitem fi natural join filmparticipation fp2 natural join
      person p
      where fi.filmtype = 'C'
      and p.gender = 'F'
      and fp2.parttype like 'cast'
      group by p.personid
      having count(fp2.partid) > 50) as p1,

     (select p.personid, fp.filmid
      from filmparticipation fp natural join person p,

      (select fpB.filmid, min(pB.lastname) as lastname
       from filmitem fi natural join filmparticipation fpB
```

```

        natural join person pB
        where fi.filmtype like 'C'
        and pB.gender like 'F'
        and fpB.parttype like 'cast'
        group by fpB.filmid) as m

where m.filmid = fp.filmid
and m.lastname = p.lastname) as p2
where p.personid = p1.personid
and p.personid = p2.personid;

```

Som sagt funker ikke denne spørringen. Tanken er i alle fall å finne to tabeller

- (i) som finner alle kvinnelige skuespillere som har spilt i mer enn 50 kinofilmer
- (ii) som finner alle kvinnelige skuespillere som dukker opp først, alfabetisk, i skuespillerlisten

og så gjøre en krysning over disse tabellene for å finne de som oppfyller oppgaveteksten.

## E

```

select p.personid , p.firstname , p.lastname
from person p natural join filmparticipation fp natural join film f
where fp.parttype = 'director'
and p.gender = 'M'
group by p.personid , p.firstname , p.lastname
having count(*) > 50
and count(*) <= (select max(c.count) from (
    select count(*)
    from person pB natural join filmparticipation fp
    where pB.gender = 'F'
    and fp.parttype like 'cast'
    and fp.filmid in (
        select f.filmid
        from person pB natural join filmparticipation fp
        natural join film f
        where pB.personid = p.personid
        #and fp.parttype = 'director' \ (med denne
        #klausulen faar vi 0 tupler!\)
    )
group by pB.personid) as c);

```