

# Andre sett obligatoriske oppgaver i INF3100 V2013

Oppgavesettet skal i utgangspunktet løses av grupper på to og to studenter som leverer felles besvarelse. Vi godkjenner også individuelle besvarelser, men oppfordrer dere altså til heller å finne en å samarbeide med. Vi godkjenner ikke grupper med mer enn to studenter. Vi vil ved behov foreta stikkprøver der vi plukker ut enkeltpersoner som må gjennomgå sin besvarelse med oss.

Gjennomføring og innlevering av oppgaven skal skje i henhold til gjeldende retningslinjer ved Institutt for informatikk, se

<http://www.mn.uio.no/ifi/studier/admin/obliger/>

*Enhver innlevering av besvarelse på en obligatorisk oppgave tas som en bekref-  
telse på at retningslinjene er lest og forstått.*

**Innleveringsfrist: Fredag 3. mai kl. 24:00.**

*Fristen er absolutt, og det blir ikke gitt utsettelse. Alle spørsmålene må besvares  
for å få godkjent besvarelsen.*

## Oppgave 1–4 Eksamen 2012

Løs oppgave 1 (FDer og normalformer), oppgave 2 (SQL), oppgave 3 (relasjonsalgebra og optimering av spørringer) og oppgave 4 (transaksjonsprotokoller) fra eksamen i INF3100 våren 2012. (Scenario: Supermarkedkjeden Rama.)

## Oppgave 5 Implementasjon

Folkeregisteret inneholder informasjon om alle innbyggere i Norge og hvor de bor. Alle personer har et entydig 11-sifret fødselsnummer **fnr**. Når noen

flytter, må de sende inn en flyttemelding til folkeregisteret. Flyttemeldingen kan omfatte mer enn én person, f.eks. hvis en hel familie flytter. Hver flyttemelding er identifisert ved et løpenummer *mid*. Forøvrig inneholder den informasjon om flyttedato, gammel og ny adresse. Anta at denne informasjonen er organisert i følgende relasjoner der primærnøkklene er understreket:

```
Person(fnr, fornavn, etternavn)
Flyttemelding(mid, flyttedato, fraadr, tiladr)
FlyttetPerson(mid, fnr)
```

Betrakt følgende SQL-spørring som finner alle flyttemeldingene (i form av løpenummer og flyttedato) til Jo Å:

```
SELECT Flyttemelding.mid, Flyttemelding.flyttedato
FROM   Person, Flyttemelding, FlyttetPerson
WHERE  Person.fnr = FlyttetPerson.fnr AND
       Flyttemelding.mid = FlyttetPerson.mid AND
       Person.fornavn = 'Jo' AND
       Person.etternavn = 'Å';
```

## 5a Parsering

- (i) Bruk den enkle grammatikken under til å lage et parseringstre for spørringen over.

Elementære syntaktiske kategorier som `<attribute>`, `<relation>` og `<pattern>` oversettes med henholdsvis navnet på attributtet, navnet på relasjonen og en streng i enkle anførselstegn.

```
<query> ::= <SFW>
```

```
<SFW> ::= SELECT <selList> FROM <fromList> WHERE <condition>
```

```
<selList> ::= <attribute>
```

```
<selList> ::= <attribute>, <selList>
```

```
<fromList> ::= <relation>
```

```
<fromList> ::= <relation>, <fromList>
```

```
<condition> ::= <condition> AND <condition>
```

```
<condition> ::= <attribute> = <attribute>
```

```

<condition> ::= <attribute> = <pattern>
<condition> ::= <attribute> LIKE <pattern>
<condition> ::= <attribute> IS NULL

```

## 5b Logisk spørreplan

- (i) Konverter parseringstreet i oppgave 5a til en logisk spørreplan i relasjonsalgebra (tegn uttrykkstreet).
- (ii) Optimer den logiske spørreplanen i punkt (i) hvis den ikke alt er på optimal form (tegn det nye uttrykkstreet).

## 5c Datalagring

Anta at vi har en disk med følgende spesifikasjoner for lagring av våre data:

Diskplater:	10 (med 2 overflater hver)
Spor:	10.000 pr. overflate
Antall sektorer pr. spor:	1000 (en ikke-sonet disk)
Byte pr. sektor:	1024
Byte pr. "gap":	64
Gjennomsnittlig søketid:	3,7 ms
Spor-spor søk:	0,3 ms
Rotasjonshastighet:	15.000 RPM

Databasen inneholder følgende antall tupler hva gjelder de to relasjonene **Person** og **Flyttmelding**:

Person:	5.000.000
Flyttmelding:	10.000.000

I tillegg gjelder følgende informasjon om diverse størrelser:

- Hver blokk har en "header" (hode) på 20 byte.
- Hver "record" (post) har et hode på 10 byte.
- Hvert attributt har følgende størrelse i antall bytes:

Person		Flyttemelding	
fnr:	4	mid:	4
fornavn:	64	flyttedato:	4
etternavn:	64	fraadr:	128
		tiladr:	128

- (i) Hva er diskens utnyttbare kapasitet?
- (ii) Hvilke faktorer inngår i å aksessere en blokk på disken, og hva er gjennomsnittlig aksestid for en vilkårlig 8 Kbyte blokk?
- (iii) Hvor stor plass trenger disse relasjonene på disken i tilfellet “unspanned” lagring (dvs. hvis ingen enkelt post er delt over flere blokker)?
- (iv) Hvor lang tid tar det å lese hele relasjonen `Person` uavbrutt hvis vi antar vilkårlig plassering av data i diskblokker på disken?
- (v) Hvis pekere er på 8 byte, hvor mange blokker må aksesserer og hva er gjennomsnittlig aksestid totalt for å finne ett tuppel (én post) i `Person` for en gitt verdi av attributtet `etternavn` hvis `person` har tett indeks på `etternavn` og indeksen er realisert ved et  $B^+$ -tre?

## Oppgave 6 Transaksjoner

Gitt tre transaksjoner  $T_1$ ,  $T_2$  og  $T_3$ :

$$\begin{aligned}
 T_1 &: r_1(A); r_1(B); w_1(C); r_1(D); \\
 T_2 &: r_2(A); r_2(B); w_2(B); w_2(C); \\
 T_3 &: r_3(A); w_3(B); r_3(D)
 \end{aligned}$$

Betrakt følgende fletting  $S$  av  $T_1$ ,  $T_2$  og  $T_3$ :

$$S : r_1(A); r_2(A); r_3(A); r_1(B); r_2(B); w_2(B); w_3(B); w_1(C); w_2(C); r_1(D); r_3(D)$$

### 6a Samtidighetskontroll, pessimistisk protokoll

An ta at vi har to typer låser – en leselås og en skrivelås. Låsene skal brukes på vanlig måte, ved at hver lese- og skriveaksjon skal ha en forutgående låseaksjon og en etterfølgende opplåsningsaksjon. Dessuten skal hver transaksjon benytte strikt tofaselåsing (strict 2PL).

La aksjonene  $sl_i(Y)$  og  $xl_i(Y)$  bety at  $T_i$  tar henholdsvis leselåsen og skrivelåsen på  $Y$ , og  $u_i(Y)$  at  $T_i$  frigir alle sine låser på  $Y$ . La aksjonene  $c_i$  og  $a_i$  bety at  $T_i$  henholdsvis committer og aborterer/må ruller tilbake.

- (i) Legg inn aksjoner av formen  $sl_i(Y)$ ,  $xl_i(Y)$  og  $u_i(Y)$  i hver av  $T_1$ ,  $T_2$  og  $T_3$  slik at de oppfyller reglene for bruk av låsene under strict 2PL. Leselåser skal frigis så snart som mulig (men i samsvar med 2PL-protokollen). Legg også inn en  $c_i$  i hver av  $T_1$ ,  $T_2$  og  $T_3$  for å tydeliggjøre hvor transaksjonene er klare til å committe.
- (ii) Beskriv hva som skjer hvis vi prøver å utføre aksjonene i de resulterende transaksjonene i henhold til strict 2PL og slik at lese/skriveaksjonene utføres mest mulig i samsvar med rekkefølgen angitt av  $S$ .

## 6b Samtidighetskontroll, optimistisk protokoll

Vi skal så se på hva som skjer hvis vi bruker Snapshot Isolation (SI).

- (i) Beskriv hva som skjer med  $T_1$ ,  $T_2$  og  $T_3$  hvis vi prøver å utføre aksjonene i transaksjonene i henhold til SI og slik at lese/skriveaksjonene utføres mest mulig i samsvar med rekkefølgen angitt av  $S$ .

## 6c Logging

Det semantiske innholdet av transaksjonen  $T_2$  består av følgende operasjoner der  $x$  og  $y$  er lokale arbeidsvariable for  $T_2$  og derfor ikke skal logges:

$$T_2 : \quad x := A; y := B; B := x + y; C := y;$$

- (i) Beskriv postene i undo/redo-loggen for transaksjonen  $T_2$  når vi initielt har verdiene  $A = 10$ ,  $B = 5$  og  $C = 0$ .
- (ii) Når skal de forskjellige typene loggposter skrives til disk?

## Slutt på obligatorisk oppgave 2