

## Løsningsforslag matoppskrifter – SQL og normalisering

### Oppgave 1

```
create table Menylinje (  
  menyid int references Meny(menyid),  
  rettnr int,  
  oppskriftsid int not null references Oppskrift(oppskriftsid),  
  drikke varchar(30),  
  primary key (menyid, rettnr),  
  unique (menyid, oppskriftsid)  
);
```

Det er unødvendig å ha **not null** på menyid og rettnr fordi dette er krav som automatisk følger av at de inngår i primærnøkkelen. **unique** angir en kandidatnøkkel, her sier den at den samme oppskriften ikke kan brukes mer enn én gang i en meny.

### Oppgave 2

Fra primærnøkkelen:

oppskriftsid → tittel, fremgangsmåte, bokid, sidenr, boktittel

Dessuten bestemmer bokid boktittelen:

bokid → boktittel

**Alternativ 1:** Hvis det er disse to som gjelder av FDer, så har vi altså (når de er skrevet ut med bare ett attributt i høyresiden):

oppskriftsid → tittel  
oppskriftsid → fremgangsmåte  
oppskriftsid → bokid  
oppskriftsid → sidenr  
oppskriftsid → boktittel  
bokid → boktittel

Eneste kandidatnøkkel er (oppskriftsid). For de fem første FDene er derfor venstresiden en supernøkkel, så de oppfyller BCNF.

I den siste FDen er bokid ikke en supernøkkel, så den bryter BCNF. Dessuten er høyresiden ikke et nøkkelattributt, så den bryter 3NF. Men venstresiden er ikke del av kandidatnøkkelen, så den oppfyller 2NF.

Totalt er derfor Oppskrift på 2NF, men bryter 3NF.

**Alternativ 2:** Vi antar i tillegg at hver bokside har nøyaktig én oppskrift, da holder FDene

bokid, sidenr → tittel  
bokid, sidenr → fremgangsmåte

Kandidatnøkkelen er som før, (oppskriftsid). På samme måte som for siste FD i alternativ 1 bryter de BCNF og 3NF, men oppfyller 2NF. Så fortsatt er Oppskrift på 2NF, men bryter 3NF.

**Alternativ 3:** Vi antar i tillegg at hver oppskrift bare er representert én gang i registeret, dvs.

bokid, sidenr → oppskriftsid

Da får vi i tilfelle kandidatnøkkelen (bokid, sidenr) i tillegg til primærnøkkelen.

Fortsatt er de fem første FDene under alternativ 1 på BCNF, for (oppskriftsid) er fortsatt en supernøkkel.

Når det gjelder den siste FDen fra alternativ 1, så er venstresiden (bokid) fortsatt ikke en supernøkkel, så den bryter BCNF. Høyresiden (boktittel) er fortsatt ikke et nøkkelattributt, så den bryter 3NF. Dessuten er venstresiden del av en kandidatnøkkel (den nye), så nå bryter den også 2NF.

De to FDene fra alternativ 2 og den nye FDen i alternativ 3 har alle venstresider som er supernøkler (den nye kandidatnøkkelen), så de oppfyller BCNF.

Men siden en av FDene bryter 2NF, er Oppskrift på 1NF, men bryter 2NF.

### Oppgave 3

Fremmednøkkelen fra Gjesteliste til Meny er (menydato).

For å håndheve fremmednøkkelen, må DBMSet

- sjekke når et nytt tuppel legges til Gjesteliste at menydatoen fins som verdi i et tuppel i Meny
- sjekke at hvis et tuppel i Gjesteliste endrer sin verdi for menydatoen at denne alt fins i et tuppel i Meny
- sjekke når et tuppel slettes i Meny at ingen av tuplene i Gjesteliste har denne menydatoen

- sjekke hvis et tuppel i Meny endrer sin menydatooverdi at ingen av tuplene i Gjesteliste har den gamle menydatoen som verdi

#### Oppgave 4

```
select oppskriftsid, tittel
from Oppskrift
where fremgangsmåte like '%stekeovn%';
```

#### Oppgave 5

```
select l.rettnr, o.tittel, l.drikke
from Meny m, Menylinje l, Oppskrift o
where m.menyid = l.menyid and l.oppskriftsid = o.oppskriftsid and
      m.menydato = '2009-06-23'
order by l.rettnr asc;
```

#### Oppgave 6

Lager likegodt et view av denne siden den trengs i oppgave 7.

```
create view AntallRetter as
select menyid, count(*) as ant
from Menylinje
group by menyid;
```

#### Oppgave 7

```
select menyid, ant
from AntallRetter
where ant >= all (select ant from AntallRetter);
```

#### Oppgave 8

Finn først ut hvor mange ganger pr. år hver gjest har vært invitert:

```
create view oversikt as
select gjest, år, count(menydato) as antall
from (select gjest, extract(year from menydato) as år, menydato
      from Gjesteliste ) as g
group by gjest, år;
```

Deretter luker vi ut de gjestene som har vært invitert to eller flere ganger ett eller annet år:

```
(select distinct gjest from Gjesteliste)
except all
(select gjest from oversikt where antall >= 2);
```

Må enten ha **distinct** i første select, da går **except all** (bagsubtraksjon) bra, eller droppe **distinct** i første select og bruke **except** (mengdesubtraksjon).

### Oppgave 9

```
select distinct ingrediens
from Ingrediensoversikt
where oppskriftsid not in (select oppskriftsid from Menylinje);
```