

Indre select-setninger
Nestede select-setninger

Setninger som beregner
en tabell med kun én verdi

```
select avg(lonn)
from Ansatt
where avd = 'marketing' ;
```

```
select avg(lonn)from Ansatt where avd = 'marketing' ;
```

```
avg
```

```
-----  
583874.58
```

```
(1 row)
```

```
select 2 * avg(lonn)
from Ansatt
where avd = 'marketing' ;
```

```
select 2*avg(lonn)from Ansatt where avd = 'marketing' ;
```

```
avg
```

```
-----  
1167749.16
```

```
(1 row)
```

Eksempel vi så sist:

Ansatt(id, navn, lønn, avd)

*Finn antall selgere som tjener mer enn det dobbelte av
markedsførernes gjennomsnittslønn*

```
select count *  
from Ansatt  
where avd = 'salg' and  
       lønn > ( select 2 * avg(lønn)  
                from Ansatt  
                where avd = 'marketing' );
```

Den indre setningen beregner en relasjon med én verdi.
Tabellen har m.a.o. én kolonne og ett tuppel. Hvis det ikke var slik,
ville >-tegnet ikke gi noen mening.

Eksempel vi så sist:

Ansatt(id, navn, lønn, avd)

*Finn antall selgere som tjener mer enn det dobbelte av
markedsførernes gjennomsnittslønn*

```
select count *  
from Ansatt  
where avd = 'salg' and  
       lønn > ( 1167749.16 );
```

Den indre setningen beregner en relasjon med én verdi.
Tabellen har m.a.o. én kolonne og ett tuppel. Hvis det ikke var slik,
ville >-tegnet ikke gi noen mening.

Eksempel vi så sist:

Ansatt(id, navn, lønn, avd)

Avdeling(avd, leder)

Prosjektplan(pnr, id, timer)

For hvert prosjekt, list opp medvirkende avdelinger som har minst 10 ansatte og som har ført minst 100 timer på prosjektet.

Sorter dem etter innsats:

```
select    pnr as prosjekt,
          avd as avdeling, sum(timer) as innsats
from      Ansatt A, Avdeling D, Prosjektplan P
where     A.avd = D.avd and A.id = P.id
group by P.pnr, D.avd
having    sum(timer) > 99 and
          9 < ( select count(*)
                from   Ansatt A1
                where  A1.avd = D.avd )
order by prosjekt, innsats desc;
```

Eksempel vi så sist:

Ansatt(id, navn, lønn, avd)

Avdeling(avd, leder)

Prosjektplan(pnr, id, timer)

Legg merke til den indre select-setningen som har en seleksjonsbetingelse som sammenligner verdier i A1.avd med verdier i attributtet D.avd fra den ytre setningen.

```
select    pnr as prosjekt,
          avd as avdeling, sum(timer) as innsats
from      Ansatt A, Avdeling D, Prosjektplan P
where     A.avd = D.avd and A.id = P.id
group by  P.pnr, D.avd
having    sum(timer) > 99 and
          9 < ( select count(*)
                from  Ansatt A1
                where  A1.avd = D.avd )
order by  prosjekt, innsats desc;
```


Eksempel vi så sist:

Ansatt(id, navn, lønn, avd)

Avdeling(avd, leder)

Prosjektplan(pnr, id, timer)

Legg merke til den indre select-setningen som har en seleksjonsbetingelse som sammenligner verdier i A1.avd med verdier i attributtet D.avd fra den ytre setningen.

```
select    pnr as prosjekt,
          avd as avdeling, sum(timer) as innsats
from      Ansatt A, Avdeling D, Prosjektplan P
where     A.avd = D.avd and A.id = P.id
group by  P.pnr, D.avd
having    sum(timer) > 99 and
          9 < ( select count(*)
                from  Ansatt A1
                where  A1.avd = D.avd )
order by  prosjekt, innsats desc;
```

Den indre
setningen blir
en tabell med
én verdi

Eksempel vi så sist:

Ansatt(id, navn, lønn, avd)

Avdeling(avd, leder)

Prosjektplan(pnr, id, timer)

Legg merke til den indre select-setningen som har en seleksjonsbetingelse som sammenligner verdier i A1.avd med verdier i attributtet D.avd fra den ytre setningen.

```
select    pnr as prosjekt,
          avd as avdeling, sum(timer) as innsats
from      Ansatt A, Avdeling D, Prosjektplan P
where     A.avd = D.avd and A.id = P.id
group by  P.pnr, D.avd
having    sum(timer) > 99 and
          9 < ( select count(*)
                from  Ansatt A1
                where  A1.avd = D.avd )
order by  prosjekt, innsats desc;
```

Den indre
setningen blir
en tabell med
én verdi

korrellert spørring

Setninger som beregner
en tabell med kun én verdi

... kan forkomme der vi kan ha en
konstant

Mest vanlig i

- where
- having

Setninger som beregner
en tabell med flere kolonner eller
tupler

... kan forkomme hvor ?

```
select    [distinct] <resultatattributter>
  from    <tabelle>
[ where   <utvalgbetingelse> ]
[ group by <grupperingsattributter>
[ having  <resultatbetingelse> ] ]
[ order by <ordningsattributter> ]
```

```
select    [distinct] <resultatattributter>
  from      <tabeller>
[ where    <utvalgbetingelse> ]
[ group by <grupperingsattributter>
[ having   <resultatbetingelse> ] ]
[ order by <ordningsattributter> ]
```

Husk at en select-setning er en relasjon (tabell) !

```
select    [distinct] <resultatattributter>
  from    T1, T2, ..., Tk
[ where   <utvalgbetingelse> ]
[ group by <grupperingsattributter>
[ having   <resultatbetingelse> ] ]
[ order by <ordningsattributter> ]
```

Husk at en select-setning er en relasjon (tabell) !


```
select      [distinct] <resultatattributter>
  from      T1,
           T2,
           ...,
           (select...) as nyTabell,
           ...,
           Tk
[ where    <utvalgbetingelse> ]
[ group by <grupperingsattributter>
[ having   <resultatbetingelse> ] ]
[ order by <ordningsattributter> ]
```

```
fdb=> \d film
```

```
Table "public.film"
```

Column	Type	Modifiers
filmid	integer	
title	text	not null
prodyear	integer	

```
select title, count(*) as ant
from Film
group by title
having count(*) > 30
```

```
fdb=> \d film
```

```
Table "public.film"
```

Column	Type	Modifiers
filmid	integer	
title	text	not null
prodyear	integer	

```
( select title, count(*) as ant  
  from Film  
  group by title  
  having count(*) > 30  
 ) as popTitler
```

```
select max(popTitler.ant)
from ( select title, count(*) as ant
      from Film
      group by title
      having count(*) > 30
      ) as popTitler
;
```

```
select max(popTitler.ant)
from ( select title, count(*) as ant
      from Film
      group by title
      having count(*) > 30
      ) as popTitler
;
```

- Den indre select-setningen blir her som en tabell med tabellnavnet PopTitler

```
select max(popTitler.ant)
from ( select title, count(*) as ant
      from Film
      group by title
      having count(*) > 30
      ) as popTitler
;
```

- Den indre select-setningen blir her som en tabell med tabellnavnet PopTitler
- Vi må alltid gi et navn til indre select-setninger i from-delen

Film(filmid, title, prodyear)

```
select F.prodyear, count(*) as ant
from Film F
  natural join
  ( select title, count(*) as ant
    from Film
    group by title
    having count(*) > 30
  ) as popTitler
group by F.prodyear
order by ant ;
```



```
fdb=> \d Filmcountry
Table "public.filmcountry"
Column | Type | Modifiers
-----+-----+-----
filmid | integer |
country | text |
```

```
Film(filmid, title, prodyear)
Filmcountry(filmid, country)
```

```
select C.country, count(*) as ant
from Film F
  natural join
  Filmcountry C
  natural join
  ( select title, count(*) as ant
    from Film
    group by title
    having count(*) > 30
  ) as T
group by C.country
having count(*) > 5
order by ant ;
```

Film(filmid, title, prodyear)
Filmcountry(filmid, country)

```
select T.title, count(*)  
from Film F  
  natural join  
    Filmcountry C  
  natural join  
    ( select title, count(*) as ant  
      from Film  
      group by title  
      having count(*) > 30  
    ) as T  
where C.country = 'Norway'  
group by T.title ;
```

Views

```
( select title, count(*) as ant  
  from Film  
  group by title  
  having count(*) > 30  
 ) as popTitler
```

```
create view PopTitler (title, ant) as
select title, count(*) as ant
from Film
group by title
having count(*) > 30
;
```

```
create view PopTitler (tittel, ant) as  
select title, count(*)  
from Film  
group by title  
having count(*) > 5  
;
```

```
create view PopTitler (tittel, ant) as
select title, count(*)
from Film
group by title
having count(*) > 5
;
```

Et view kan brukes som en tabell, gjerne i stedet for en indre select-setning. I dette viewet er det 2 attributter: **tittel** og **ant**


```
create view PopTitrer as
select title, count(*)
from Film
group by title
having count(*) > 5
;
```

Et view kan brukes som en tabell, gjerne i stedet for en indre select-setning. I dette viewet er det 2 attributter: **title** og **count**

```
create view PopTitrer as
select title as tittel, count(*) as ant
from Film
group by title
having count(*) > 5
;
```

Et view kan brukes som en tabell, gjerne i stedet for en indre select-setning. I dette viewet er det 2 attributter: **tittel** og **ant**

```
fdb=> \d filmgenre
      Table "public.filmgenre"
  Column |      Type      | Modifiers
-----+-----+-----
  filmid | integer        |
  genre  | text           |
```

```
select P.tittel, count(*)  
from Film F  
    natural join  
        Filmgenre G  
    join  
        Poptitler P  
    on F.title = P.tittel  
where G.genre = 'Drama'  
group by P.tittel ;
```

```
select P.tittel, count(*)  
from Film F  
    natural join  
        Filmgenre G  
    join  
        Poptitler P  
    on F.title = P.tittel  
where G.genre = 'Drama'  
group by P.tittel ;
```

```
create view PopTitler as  
select title as tittel, count(*) as ant  
from Film  
group by title  
having count(*) > 5  
;
```

```
select P.title, count(*)  
from Film F  
    natural join  
    Filmgenre G  
    natural join  
    Poptitler P  
  
where G.genre = 'Drama'  
group by P.title ;
```

```
create view PopTitler as  
select title as title, count(*) as ant  
from Film  
group by title  
having count(*) > 5  
;
```

```
select P.title, count(*)  
from Film F
```

```
  natural join
```

```
    Filmgenre G
```

```
  natural join
```

```
    Poptitler P
```

```
where G.genre = 'Drama'  
group by P.title ;
```

```
create view PopTitler as  
select title as title, count(*) as ant  
from Film  
group by title  
having count(*) > 5  
;
```

```
select P.title, count(*)  
from Film F  
    natural join  
    Filmgenre G  
    natural join
```

```
create view PopTitler as  
select title as title, count(*) as ant  
from Film  
group by title  
having count(*) > 5  
;
```

```
    Poptitler P  
where G.genre = 'Drama'  
group by P.title ;
```



```
select P.title, count(*)  
from Film F  
  natural join  
    Filmgenre G  
  natural join
```

```
( select title as title, count(*) as ant  
  from Film  
  group by title  
  having count(*) > 5 )
```

```
  Poptitler P  
where G.genre = 'Drama'  
group by P.title ;
```

```
select P.title, count(*)  
from Film F
```

```
  natural join
```

```
    Filmgenre G
```

```
  natural join
```

```
( select title as title, count(*) as ant  
  from Film  
  group by title  
  having count(*) > 5 ) as P
```

```
where G.genre = 'Drama'  
group by P.title ;
```

```
select P.tittel, count(*) as ant
from Film F
  join
    Poptitler P
  on F.title = P.tittel

group by P.tittel , ant
having ant = ( select max(ant)
              from PopTitler )
;
```

```
create view PopTitler as
select title as tittel, count(*) as ant
from Film
group by title
having count(*) > 5
;
```

```
select P.tittel, count(*) as ant
from Film F
  join
    Poptitler P
  on F.title = P.tittel
```

```
group by P.tittel , ant
having ant = ( select max(ant)
              from PopTitler )
```

Indre selectsetning
som beregner én
verdi

;

```
create view PopTitler as
select title as tittel, count(*) as ant
from Film
group by title
having count(*) > 5
;
```

```
select P.tittel, count(*) as ant
from Film F
  join
    Poptitler P
  on F.title = P.tittel
```

```
group by P.tittel , ant
having ant = ( select max(ant)
              from PopTitler )
;
```

Indre selectsetning
som beregner én
verdi

*her kan vi igjen erstatte
viewet PopTitler med
select-setningen som definerer
det*

```
create view PopTitler as
select title as tittel, count(*) as ant
from Film
group by title
having count(*) > 5
;
```

```
select P.tittel, count(*) as ant
from Film F
  join
    Poptitler P
  on F.title = P.tittel
```

```
group by P.tittel , ant
having ant = ( select max(ant)
              from PopTitler )
;
```

Indre selectsetning
som beregner én
verdi

*her kan vi igjen erstatte
viewet PopTitler med
select-setningen som definerer
det*

```
create view PopTitler as
select title as tittel, count(*) as ant
from Film
group by title
having count(*) > 5
;
```

Relasjonssammenligninger

SQL har fem operatører som sammenligner med innholdet i en hel relasjon:

<i>i SQL-2</i>	<i>betyr</i>
exists R	at R har minst én forekomst
not exists R	at R ikke har noen forekomster
in R	$\in R$
not in R	$\notin R$
any R	en vilkårlig verdi i R
all R	alle verdier i R

Relasjonssammenligninger

SQL-uttrykket evalueres til **true** hvis

$e \text{ in } R$ e finnes som forekomst i R

$\text{exists } R$ R har minst én forekomst

$e = \text{any } R$ $e =$ minst én vilkårlig verdi i R

$e > \text{all } R$ $e >$ alle verdier i R

Relasjonssammenligninger

SQL-uttrykket evalueres til **true** hviss

$e \text{ in } R$ e finnes som forekomst i R

$\text{exists } R$ R har minst én forekomst

$e = \text{any } R$ $e =$ minst én vilkårlig verdi i R

$e > \text{all } R$ $e >$ alle verdier i R

Kan forekomme i where- og having-betingelser

Ansatt(id, navn, lønn, avd)

Finn antall selgere som tjener mer enn samtlige markedsførere

Ansatt(id, navn, lønn, avd)

```
select count(*)  
from   Ansatt  
where  avd = 'salg' and  
       lønn > all (select lønn  
                  from   Ansatt  
                  where  avd = 'marketing');
```

Finn antall selgere som tjener mer enn samtlige markedsførere

Ansatt(id, navn, lønn, avd)

```
select count(*)
from   Ansatt
where  avd = 'salg' and
       lønn > all (select lønn
                  from   Ansatt
                  where  avd = 'marketing');
```

Den indre setningen gir verdien I attributtet lønn for alle ansatte som har 'marketing' som verdi I attributtet avd

Finn antall selgere som tjener mer enn samtlige markedsførere

Film(filmid, title, prodyear)

Filmparticipation(partid, personid, filmid, parttype)

Filmer og rolle for personer med navnet Michael King

Film(filmid, title, prodyear)

Filmparticipation(partid, personid, filmid, parttype)

Filmer og rolle for personer med navnet Michael King

```
select personid, title, parttype
from      film f
natural join filmparticipation fp
natural join person p
where personid = any ( select personid
                       from Person
                       where lastname like 'King'
                       and firstname like 'Michael' );
```

Film(filmid, title, prodyear)

Filmparticipation(partid, personid, filmid, parttype)

Filmer og rolle for personer med navnet Michael King

```
select personid, title, parttype
from      film f
natural join filmparticipation fp
natural join person p
where personid = any ( select personid
                       from Person
                       where lastname like 'King'
                       and firstname like 'Michael' );
```

Den indre setningen gir alle forekomster av personid fra Filmparticipation-tabellen med navnet Michael King.

Film(filmid, title, prodyear)
Filmcountry(filmid, country)

Finn antall filmer som er produsert i nordiske land ...

Film(filmid, title, prodyear)
Filmcountry(filmid, country)

```
select C.country, count(*)  
from   Film F  
       natural join  
       Filmcountry C  
where  C.country in ('Norway', 'Sweden', 'Denmark',  
                    'Finland', 'Iceland')  
group by C.country;
```

Ansatt(id, navn, lønn, avd)
Avdeling(avd, leder)
Prosjektplan(pnr, id, timer)

navn på ansatte som ikke har ført noen prosjekttimer

Ansatt(id, navn, lønn, avd)
Avdeling(avd, leder)
Prosjektplan(pnr, id, timer)

```
select navn  
from   Ansatt  
where  id not in (select id  
                  from   Prosjektplan);
```

navn på ansatte som ikke har ført noen prosjekttimer

Ansatt(id, navn, lønn, avd)
Avdeling(avd, leder)
Prosjektplan(pnr, id, timer)

```
select navn  
from   Ansatt  
where  id not in (select id  
                  from   Prosjektplan);
```

Den indre setningen gir *alle*
ansatt-id som finnes i Prosjektplan

navn på ansatte som ikke har ført noen prosjekttimer

Ansatt(id, navn, lønn, avd)
Avdeling(avd, leder)
Prosjektplan(pnr, id, timer)

*Finn navn på ansatte som skal arbeide mer enn 10 timer
på samtlige av sine prosjekter*

Ansatt(id, navn, lønn, avd)
Avdeling(avd, leder)
Prosjektplan(pnr, id, timer)

```
select A.navn  
from   Ansatt A  
where  not exists ( select *  
                   from   Prosjektplan P  
                   where  P.id = A.id  
                   and P.timer <= 10 );
```

*Finn navn på ansatte som skal arbeide mer enn 10 timer
på samtlige av sine prosjekter*

Ansatt(id, navn, lønn, avd)
Avdeling(avd, leder)
Prosjektplan(pnr, id, timer)

```
select A.navn
from   Ansatt A
where  not exists ( select *
                   from   Prosjektplan P
                   where  P.id = A.id
                   and P.timer <= 10 );
```

Den indre setningen gir *alle* forekomster av
prosjektplantabellen der den ansatte A.id
skal jobbe 10 timer eller mindre på
prosjektet P.pnr

*Finn navn på ansatte som skal arbeide mer enn 10 timer
på samtlige av sine prosjekter*

Ansatt(id, navn, lønn, avd)

Avdeling(avd, leder)

Prosjekt(pnr, leder)

Prosjektplan(pnr, id, timer)

*Finn navn på de ansatte som ikke skal delta på noe
prosjekt ledet av en avdelingsleder*

Ansatt(id, navn, lønn, avd)
Avdeling(avd, leder)
Prosjekt(pnr, leder)
Prosjektplan(pnr, id, timer)

*Finn navn på de ansatte som ikke skal delta på noe
prosjekt ledet av en avdelingsleder*

```
select navn
from   Ansatt A
where  not exists (
        select *
        from   Prosjekt P, Prosjektplan PL
        where  PL.id = A.id and
               PL.pnr = P.pnr and
               P.leder in (select leder
                           from   Avdeling)
      );
```

Ansatt(id, navn, lønn, avd)

Avdeling(avd, leder)

Prosjektplan(pnr, id, timer)

Finn navn på ansatte som skal delta på alle prosjekter

Ansatt(id, navn, lønn, avd)
Avdeling(avd, leder)
Prosjektplan(pnr, id, timer)

Finn navn på ansatte som skal delta på alle prosjekter

```
select A.navn
from   Ansatt A
where  not exists (
        select P1.pnr
        from   Prosjektplan P1
        where  not exists (
                select *
                from   Prosjektplan P2
                where  P2.pnr = P1.pnr
                and    P2.id = A.id
            )
    );
```

```
select count(filmid)
from filmcountry
where
    filmid not in (select filmid from film)
;
```

```
select count(filmid)
from filmcountry
where
    filmid not in (select filmid from film)
;
```

```
fdb=> select count(filmid)
fdb-> from filmcountry
fdb-> where
fdb-> filmid not in (select filmid from film)
fdb-> ; -- denne spørringen hadde eksekvert i mer
      -- enn 3 timer da jeg stanset den:
Cancel request sent
ERROR: canceling statement due to user request
fdb=>
```

```
select count(filmid)
from filmcountry f
where
    not exists (select filmid from film ff
                where ff.filmid = f.filmid)
;
```

```
-- tok mindre enn 1 - ett - sekund !
```

```
select count(filmid)
from filmcountry f
where
    not exists (select filmid from film ff
                where ff.filmid = f.filmid)
;
```

-- tok mindre enn 1 - ett - sekund !

```
select count(filmid)
from filmcountry
where
    filmid not in (select filmid from film)
;
```