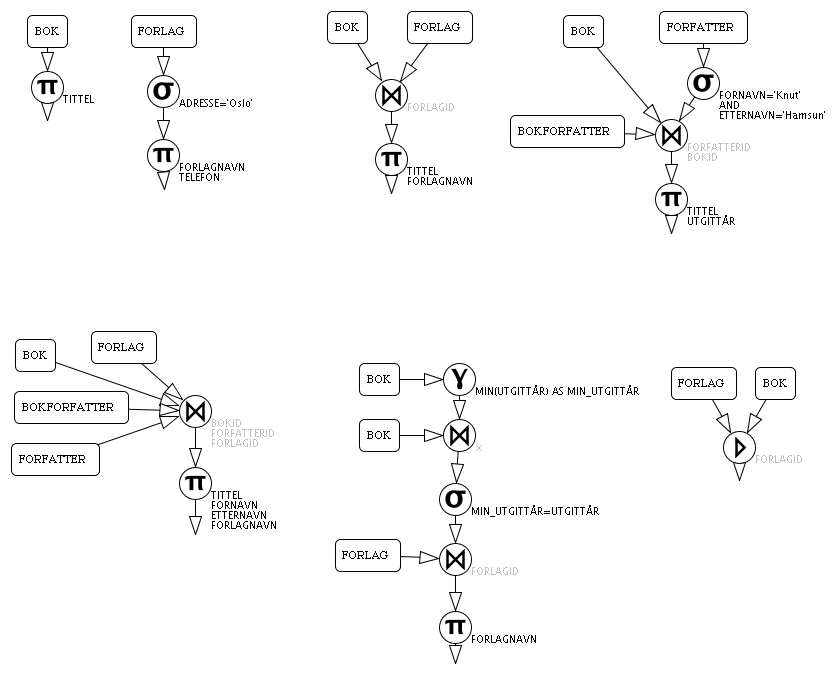
Databaser Øving 3 – Arve Nygård

# Oppgave 1:



# Oppgave 2:

*-- -----------------------------------------------------*

*-- Table `Poststed`*

*-- -----------------------------------------------------*

**CREATE** **TABLE** IF **NOT** **EXISTS** `Poststed` (

`postnr` INT **NOT** **NULL** ,

`poststed` VARCHAR(45) **NULL** ,

**PRIMARY** **KEY** (`postnr`) ,

**UNIQUE** **INDEX** `postnr\_UNIQUE` (`postnr` **ASC**) )

*-- -----------------------------------------------------*

*-- Table `Kunde`*

*-- -----------------------------------------------------*

**CREATE** **TABLE** IF **NOT** **EXISTS** `Kunde` (

`kundenr` INT **NOT** **NULL** AUTO\_INCREMENT ,

`navn` VARCHAR(45) **NOT** **NULL** ,

`kredittgrense` INT **NOT** **NULL** **COMMENT** '\n ' ,

`postnr` INT **NOT** **NULL** ,

**PRIMARY** **KEY** (`kundenr`) ,

**INDEX** `kunde\_poststed\_idx` (`postnr` **ASC**) ,

**CONSTRAINT** `kunde\_poststed`

**FOREIGN** **KEY** (`postnr` )

**REFERENCES** `Poststed` (`postnr` )

**ON** **DELETE CASCADE**

**ON** **UPDATE** **CASCADE**)

*-- -----------------------------------------------------*

*-- Table `Artikkel`*

*-- -----------------------------------------------------*

**CREATE** **TABLE** IF **NOT** **EXISTS** `Artikkel` (

`artnr` INT **NOT** **NULL** AUTO\_INCREMENT ,

`navn` VARCHAR(45) **NOT** **NULL** ,

`ant` INT **NOT** **NULL** ,

`pris` INT UNSIGNED **NOT** **NULL** ,

**PRIMARY** **KEY** (`artnr`) )

*-- -----------------------------------------------------*

*-- Table `Bestilling`*

*-- -----------------------------------------------------*

**CREATE** **TABLE** IF **NOT** **EXISTS** `Bestilling` (

`artnr` INT **NOT** **NULL** ,

`kundenr` INT **NOT** **NULL** ,

`kvantum` INT UNSIGNED **NOT** **NULL** ,

**PRIMARY** **KEY** (`artnr`, `kundenr`) ,

**INDEX** `artnr\_idx` (`artnr` **ASC**) ,

**INDEX** `kundnr\_idx` (`kundenr` **ASC**) ,

**CONSTRAINT** `artnr`

**FOREIGN** **KEY** (`artnr` )

**REFERENCES** `Artikkel` (`artnr` )

**ON** **DELETE** **CASCADE**

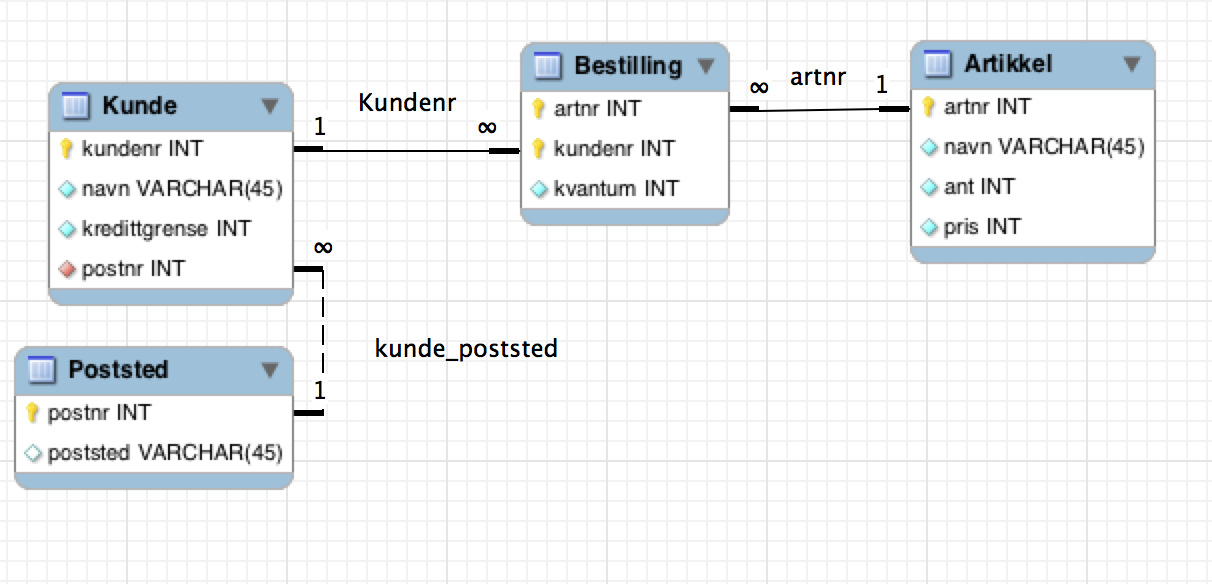
**ON** **UPDATE** **CASCADE**,

**CONSTRAINT** `kundnr`

**FOREIGN** **KEY** (`kundenr` )

**REFERENCES** `Kunde` (`kundenr` )

**ON** **DELETE** **CASCADE**

**ON** **UPDATE** **CASCADE**) 

# Oppgave 3

-- a)

select tittel from bok;

-- b)

select \* from forfatter where nasjoforfatternalitet like "Norsk";

-- c)

select forlagnavn, telefon from forlag where adresse like "Oslo";

-- e)

SELECT b.tittel, b.utgittår

FROM forfatter f

LEFT JOIN bokforfatter bf ON bf.forfatterid = f.forfatterid

LEFT JOIN bok b ON bf.bokid = b.bokid

WHERE

fornavn like "Knut" and

etternavn like "hamsun";

-- f)

select fornavn, etternavn, fødeår

from forfatter

where etternavn like 'H%';

-- g)

select count(\*) from forlag;

-- i)

select

f.fornavn,

f.etternavn,

count(\*) as 'antall'

from forfatter f

left join bokforfatter bf on f.forfatterid = bf.forfatterid

left join bok b on b.bokid = bf.bokid

group by f.fornavn, f.etternavn

order by antall desc;

-- j)

select tittel, utgittår from bok order by utgittår asc limit 1;

-- k)

select f.forlagnavn, count(\*) as 'antall' from forlag f

left join bok b on f.forlagid = b.forlagid

group by f.forlagid

having antall > 2;

-- l)

select forlagnavn from forlag f

left join bok b on b.forlagid = f.forlagid

where b.forlagid IS NULL;

# Oppgave 4

1. Hensikten med views er å spare tid, samt å kunne begrense tilgang. Grunnen til at det kan oppstå problemer med inserts mot et view, er at et view kan stamme fra flere tabeller, og det kan være ambiguous kolonnenavn, eller constraints kan være brutt (inconsistent data). I tillegg vil det bli umulig å håndtere aggeregates.
2. create view lollert (pname, dname, num\_employees, total\_hours) as  
   select  
    pname,  
    dname,  
    count(\*),  
    sum(hours)  
    from  
    project,  
    department,  
    works\_on  
   where  
    dnum = dnumber and  
    pnumber = pno  
   group by pname
3. 1 og 2 er gyldige. Oppdateringene er ikke lovlige da det er brukt aggeregates.   
   Siden viewet bare er basert på én tabell, blir spørringene bare en query mot tabellen med et utvalg av kolonner.

# Oppgave 5

-- a)

select

sno,

sname

from supplier

where status > 15;

-- b)

select s.sname, s.city from supplier s

left join SuppliesPart sp on s.sno = sp.sno

left join Part p on sp.pno = p.pno

where p.pname like 'Screw';

-- c)

select p.pno, p.pname, count(\*) as 'ant' from part

left join SuppliesPart sp on p.pno = sp.pno

group by p.pno

having ant>1;

-- d)  
select count(\*) from supplier;

-- e)  
select s.city from supplier s

left join SuppliesPart sp on s.sno=sp.sno  
left join Part p on sp.pno = p.pno  
group by s.sno  
having max(p.weight) > 10

-- f)  
select distinct sname from supplier   
where sno not in   
(

select sp.sno

from SuppliesPart sp, Part p  
where sp.pno=p.pno and   
p.pname='Screw'

)

order by sname;