

IN2090 LF uke 9

1: filmid, navn og produksjonsår for filmer som inneholder ordet «Norge»

```
SELECT filmid, title, prodyear
FROM film
WHERE title LIKE '%Norge%';
```

2: Hvilke verdier forekommer i attributtet filmtype i relasjonen filmitem? Lag en oversikt over filmtypene og hvor mange filmer innen hver type.

```
SELECT filmtype, COUNT(*) ant
FROM filmitem
GROUP BY filmtype
ORDER BY ant DESC;
```

3 filmid for kinofilmer som har filmtittelen Love (kinofilmer har filmtype «C»)

```
SELECT fi.filmid
FROM filmitem fi INNER JOIN film f ON f.filmid = fi.filmid
WHERE fi.filmtype = 'C'
AND f.title = 'Love';
```

4. Mange titler har vært brukt i flere filmer. Skriv ut en oversikt over titler som har vært brukt i mer enn 30 filmer. Bak hver tittel skriv antall ganger den er brukt. Ordne linjene med hyppigst forekommende tittel først. (12 eller 26)

```
SELECT title, COUNT(*) AS ant
FROM film
GROUP BY title
HAVING COUNT(*) > 30
ORDER BY ant DESC;
-- Bare kinofilmer (12 rader)
SELECT title, COUNT(*) AS ant
FROM film INNER JOIN filmitem ON film.filmid = filmitem.filmid
WHERE filmitem.filmtype = 'C'
GROUP BY title
HAVING COUNT(*) > 30
ORDER BY ant DESC;
```

5. Hva er gjennomsnittlig score (rank) for filmer med over 100 000 stemmer (votes)?

```
select avg(rank)
from filmrating
where votes > 100000;
```

6. Hvilke filmer (tittel og score) med over 100 000 stemmer har en høyere score enn snittet blant filmer med over 100 000 stemmer (subspørring!)

```
select title, rank
from film inner join filmrating on film.filmid = filmrating.filmid
where votes > 100000 and rank >= (
    select avg(rank)
    from filmrating
    where votes > 100000
);
```

7. Hvilke 100 verdier (fornavn) forekommer hyppigst i firstname-attributtet i tabellen Person?

```
SELECT firstname, COUNT(*) AS sammeFornavn
FROM Person
WHERE firstname <> ''
GROUP BY firstname
ORDER BY sammeFornavn DESC
LIMIT 100;
```

8. Hvilke to fornavn forekommer mer enn 6000 ganger og akkurat like mange ganger? (Paul og Peter, vanskelig!)

```
SELECT A.fornavn, A.antall, B.fornavn, B.antall
FROM (
    SELECT firstname fornavn, COUNT(*) antall
    FROM Person
    GROUP BY firstname
    HAVING COUNT(*) > 5999) AS A
INNER JOIN (
    SELECT firstname fornavn, COUNT(*) antall,
    FROM Person
    GROUP BY firstname
    HAVING COUNT(*) > 5999) AS B
ON A.antall = B.antall AND A.fornavn <> B.fornavn;
```

Forskjellige typer join

9. Navn på alle skuespillere (cast) i filmen, deres rolle (parttype) i filmen «Harry Potter and the Goblet of Fire» (vær presis med staving), få med tittelen til filmen også.

```
SELECT DISTINCT p.firstname, p.lastname, fp.parttype, f.title
FROM Person p NATURAL JOIN Film participation fp NATURAL JOIN film f
WHERE title = 'Harry Potter and the Goblet of Fire' AND parttype = 'cast';
```

10a. Finn fornavn og etternavn på alle personer som har deltatt i TV-serien South Park. Med **INNER JOIN**:

```
SELECT DISTINCT p.personid, p.lastname, p.firstname, s.maintitle
FROM Person p
INNER JOIN Film participation fp
    ON p.personid = fp.personid
INNER JOIN Series s
    ON s.seriesid = fp.filmid
WHERE s.maintitle = 'South Park';
```

10b. Med *implisitt join*:

```
SELECT DISTINCT p.personid, p.lastname, p.firstname, s.maintitle
FROM Person p, Film participation fp, Series s
WHERE s.seriesid = fp.filmid
    AND p.personid = fp.personid
    AND maintitle = 'South Park';
```

10c. Med **NATURAL JOIN** (gir ikke riktig resultat!)

```
SELECT DISTINCT p.personid, p.lastname, p.firstname, s.maintitle
FROM Person p NATURAL JOIN Film participation fp NATURAL JOIN Series s
WHERE s.maintitle LIKE 'South Park';
```

10d. **NATURAL JOIN** joiner «automatisk» på attributtene som har *samme navn*. Dette fungerer i join-operasjonen mellom tabellen **Person** og **Film participation** fordi begge har attributtet **personid** som det joines på. Men mellom **Film participation** og **Series** er det ingen attributter med felles navn: vi må joine på **film participation.filmid** og **series.seriesid**, og det går ikke med **NATURAL JOIN**. Vi må derfor bruke en annen join-metode (som i 10a eller 10b).

Norske filmer

11. Hvor mange filmer i filmdatabasen er norske?

```
SELECT COUNT(*) AS antallNorskeFilmer
FROM Filmcountry
WHERE country = 'Norway';
```

12. Skriv ut tittel og regissør for norske filmer produsert før 1960.

```
select film.title, person.firstname || ' ' || person.lastname as fullname
from filmcountry
  natural join film
  natural join filmparticipation
  natural join person
where filmcountry.country = 'Norway'
  and parttype = 'director'
  and prodyear < 1960;
```

13. Hvor mange filmer har Tancred Ibsen regissert?

```
SELECT COUNT(DISTINCT filmid) AS tancredIbsenFilmer
FROM Filmparticipation
  NATURAL JOIN Person
WHERE
  lastname = 'Ibsen' AND firstname = 'Tancred'
  AND parttype = 'director';
```

14. Lag en oversikt over regissører som har regissert mer enn 5 norske filmer.

```
SELECT lastname || ', ' || firstname AS navn
FROM Filmcountry
  NATURAL JOIN Film
  NATURAL JOIN Filmparticipation
  NATURAL JOIN Person
WHERE country = 'Norway'
  AND parttype = 'director'
GROUP BY lastname, firstname
HAVING COUNT(*) > 5
ORDER BY lastname ASC;
```

15. Lag en oversikt (filmtittel) over norske filmer med mer enn én regissør.

```
SELECT filmid, title
FROM Filmcountry
  NATURAL JOIN Film
  NATURAL JOIN Filmparticipation
  NATURAL JOIN Person
WHERE country = 'Norway'
  AND parttype = 'director'
GROUP BY filmid, title
HAVING COUNT(*) > 1;
```

16. Regissører som har regissert alene mer enn 5 norske filmer (utfordring!)

```
SELECT lastname || ', ' || firstname AS navn, COUNT(*) AS antall
FROM Filmcountry
  NATURAL JOIN Film
  NATURAL JOIN Filmparticipation
  NATURAL JOIN Person
WHERE country = 'Norway'
  AND parttype = 'director'
  AND filmid NOT IN
  ( SELECT filmid
    FROM Filmcountry
      NATURAL JOIN Film
      NATURAL JOIN Filmparticipation
      NATURAL JOIN Person
    WHERE country = 'Norway'
      AND parttype = 'director'
    GROUP BY filmid, title
    HAVING COUNT(*) > 1 )
GROUP BY lastname, firstname
HAVING COUNT(*) > 5
ORDER BY antall DESC;
```

Diverse oppgaver

17. tittel, produksjonsår og filmtype for alle kinofilmer som ble produsert i året 1893

```
SELECT f.title, f.prodyear, fi.filmtype
FROM film f NATURAL JOIN filmitem fi
WHERE f.prodyear = 1893;
```

18. navn på alle skuespillere (cast) i filmen Baile Perfumado

```
SELECT DISTINCT p.firstname || ' ' || p.lastname AS name
FROM film f NATURAL JOIN filmparticipation fp NATURAL JOIN person p
WHERE fp.parttype LIKE 'cast'
AND f.title LIKE 'Baile Perfumado';
```

19. tittel og produksjonsår for alle filmene som Ingmar Bergman har vært regissør (director) for. Sorter tuplene kronologisk etter produksjonsår.

```
SELECT f.title, f.prodyear
FROM film f NATURAL JOIN filmparticipation fp NATURAL JOIN person p
WHERE p.lastname LIKE 'Bergman'
AND p.firstname LIKE 'Ingmar'
AND fp.parttype LIKE 'director'
ORDER BY f.prodyear DESC;
```

20. produksjonsår for første og siste film Ingmar Bergman regisserte

```
SELECT MIN(f.prodyear) AS first, MAX(f.prodyear) AS last
FROM film f NATURAL JOIN filmparticipation fp NATURAL JOIN person p
WHERE fp.parttype LIKE 'director'
AND p.lastname LIKE 'Bergman'
AND p.firstname LIKE 'Ingmar';
```

21. tittel og produksjonsår for de filmene hvor mer enn 300 personer har deltatt, uansett hvilken funksjon de har hatt (3).

```
SELECT f.title, f.prodyear, COUNT(*) AS participants
FROM film f NATURAL JOIN filmparticipation fp
GROUP BY f.title, f.prodyear
HAVING COUNT(DISTINCT fp.personid) > 300
ORDER BY participants DESC;
-- Antatt her at vi ikke trenger å filtrere bort VG delen av filmtypen.
-- (GTA: San Andreas er jo tross alt et videospill og ikke en film)
```

22. oversikt over regissører som har regissert kinofilmer over et stort tidsspenn. I tillegg til navn, ta med antall kinofilmer og første og siste år (prodyear) personen hadde regi. Skriv ut alle som har et tidsintervall på mer enn 49 år mellom første og siste film og sorter dem etter lengden på dette tidsintervallet, de lengste først.

```
SELECT p.firstname || ' ' || p.lastname AS name, COUNT(*) AS produced,
       MIN(f.prodyear) AS first, MAX(f.prodyear) AS last,
       MAX(f.prodyear) - MIN(f.prodyear) AS periode
FROM film f NATURAL JOIN filmparticipation fp NATURAL JOIN person p
WHERE fp.parttype LIKE 'director'
GROUP BY p.personid, name
HAVING (MAX(f.prodyear) - MIN(f.prodyear) > 49)
ORDER BY periode DESC;
-- Trenger å gruppere på personid i tillegg siden to regissører kan ha samme navn.
```

23. filmid, tittel og antall medregissører (parttype 'director') (0 der han har regissert alene) for filmer som Ingmar Bergman har regissert.

```
WITH ingmarbergmanmovies AS (
  SELECT fp.filmid
  FROM filmparticipation fp INNER JOIN person p ON fp.personid = p.personid
  WHERE fp.parttype = 'director'
  AND p.firstname = 'Ingmar'
  AND p.lastname = 'Bergman'
),
ant_regissorer AS (
  SELECT fp.filmid, COUNT(*) ant
  FROM filmparticipation fp
  WHERE fp.filmid IN (SELECT * FROM ingmarbergmanmovies)
  AND fp.parttype = 'director'
  GROUP BY fp.filmid
)
SELECT f.filmid, f.title, (ar.ant - 1) AS ant_medregissorer
FROM film f INNER JOIN ant_regissorer ar ON f.filmid = ar.filmid;
```

24. filmid, antall involverte personer, produksjonsår og rating for alle filmer som Ingmar Bergman har regissert. Ordne kronologisk etter produksjonsår.

```
WITH ingmarbergmanmovies AS (  
  SELECT fp.filmid  
  FROM filmparticipation fp INNER JOIN person p ON fp.personid = p.personid  
  WHERE fp.parttype = 'director'  
  AND p.firstname = 'Ingmar'  
  AND p.lastname = 'Bergman'  
)  
crew AS (  
  SELECT fp.filmid, COUNT(*) as ant  
  FROM filmparticipation fp  
  WHERE fp.filmid IN (SELECT * FROM ingmarbergmanmovies)  
  GROUP BY filmid  
)  
SELECT f.filmid, c.ant, f.prodyear, fr.rank AS rating  
FROM film f INNER JOIN crew c ON f.filmid = c.filmid  
  INNER JOIN filmrating fr ON fr.filmid = f.filmid  
WHERE f.filmid IN (SELECT * FROM ingmarbergmanmovies)  
ORDER BY f.prodyear;
```

25. produksjonsår og tittel for alle filmer som både Angelina Jolie og Antonio Banderas har deltatt i sammen.

```
-- Kan også løses f.eks. med join og subspørring eller IN og subspørring  
select f.title, f.prodyear  
from film f natural join filmparticipation fp natural join person p  
where p.firstname = 'Angelina' and p.lastname = 'Jolie'  
  and exists (  
    select *  
    from filmparticipation fp2 natural join person p  
    where fp2.filmid = fp.filmid  
      and p.firstname = 'Antonio' and p.lastname = 'Banderas'  
  );
```


26. tittel, navn og roller for personer som har hatt mer enn én rolle i samme film blant kinofilmer som ble produsert i 2003.

```
-- (Med forskjellige roller mener vi cast, director, producer osv.
-- Vi skal altså ikke ha med de som har to ulike cast-roller)
select distinct f.title, concat(p.firstname, ' ', p.lastname), fp.parttype
from film f natural join filmparticipation fp natural join person p
inner join (
  select fp.personid, fp.filmid
  from filmparticipation fp natural join film natural join filmitem
  where film.prodyear = 2003 and filmitem.filmtype = 'C'
  group by fp.personid, fp.filmid
  having count(distinct parttype) > 1
) q on q.filmid = fp.filmid and q.personid = fp.personid
order by firstname, lastname asc;

-- eller
SELECT DISTINCT f.title, p.firstname || ' ' || p.lastname as name, fp.parttype
FROM film f NATURAL JOIN filmitem fi NATURAL JOIN filmparticipation fp NATURAL
JOIN person p
WHERE f.prodyear = 2003
AND fi.filmtype = 'C'
GROUP BY f.title, p.firstname, p.lastname, fp.parttype, fp.personid, f.filmid
HAVING (
  SELECT count(distinct fp1.parttype)
  FROM filmparticipation fp1
  WHERE fp1.personid = fp.personid AND f.filmid = fp1.filmid) > 1
ORDER BY f.title, name, fp.parttype;
```

27. navn og antall filmer for personer som har deltatt i mer enn 15 filmer i 2008, 2009 eller 2010, men som ikke har deltatt i noen filmer i 2005.

```
--Løsning med NOT EXISTS
SELECT p.firstname, p.lastname, count(distinct f.filmid)
FROM film f NATURAL JOIN filmparticipation fp NATURAL JOIN person p
WHERE (prodyear BETWEEN 2008 and 2010)
AND NOT EXISTS (
    SELECT * FROM film f1 NATURAL JOIN filmparticipation fp1
    WHERE fp.personid = fp1.personid
    AND f1.prodyear = 2005)
GROUP BY p.personid, p.firstname, p.lastname
HAVING count(distinct f.filmid)>15;

--Løsning med LEFT OUTER JOIN
SELECT p.firstname, p.lastname, count(distinct f.filmid)
FROM film f NATURAL JOIN filmparticipation fp NATURAL JOIN person p LEFT OUTER
JOIN (
    SELECT * FROM film f1 NATURAL JOIN filmparticipation fp1
    WHERE f1.prodyear = 2005) t1 on p.personid = t1.personid
WHERE (f.prodyear BETWEEN 2008 and 2010) AND t1.personid IS NULL
GROUP BY p.personid, p.firstname, p.lastname
HAVING count(distinct f.filmid)>15;

--Løsning med NOT IN
select p.firstname || ' ' || p.lastname as name, count(distinct filmid) as antall
from film f natural join filmparticipation fp natural join person p
where f.prodyear in (2008,2009,2010) and fp.personid not in (
    select personid
    from filmparticipation fp natural join film f
    where f.prodyear = 2005
)
group by fp.personid, name
having count(distinct filmid) > 15;
```

28. navn på regissør og filmtittel for filmer hvor mer enn 200 personer har deltatt, uansett hvilken funksjon de har hatt. Ta ikke med filmer som har hatt flere (mer enn én) regissører.

```
SELECT p.firstname, p.lastname, f.title
FROM film f NATURAL JOIN filmparticipation fp NATURAL JOIN person p
WHERE fp.parttype = 'director' AND
f.filmid IN(SELECT f.filmid
            FROM film f NATURAL JOIN filmparticipation fp
            GROUP BY f.filmid
            HAVING count(distinct fp.personid) > 200) AND
f.filmid not IN(SELECT f.filmid
                FROM film f NATURAL JOIN filmparticipation fp
                WHERE fp.parttype='director'
                GROUP BY f.filmid
                HAVING count(fp.parttype) > 1);

--eller
select p.firstname || ' ' || p.lastname as name, f.title
from film f natural join filmparticipation fp natural join person p
inner join (
  select filmid
  from filmparticipation
  where parttype = 'director'
  and filmid in (
    select filmid
    from filmparticipation fp
    group by filmid
    having count(*) > 200
  )
  group by filmid
  having count(*) = 1
) q on q.filmid = fp.filmid
where parttype = 'director';
```

29. navn i leksikografisk orden på regissører som har regissert alene kino- filmer med mer enn 150 deltakere og som har en regissørkarriere (jf. spørsmål 19) på mer enn 49 år.

```
select p.lastname || ', ' || p.firstname as name
from person p
where p.personid in (
  select fp.personid
  from filmparticipation fp
  where parttype = 'director'
  and fp.filmid in (
    select filmid
    from filmparticipation natural join film f
    where parttype = 'director'
    and filmid in (
      select filmid
      from filmparticipation fp
      group by filmid
      having count(*) > 150
    )
  )
  group by filmid
  having count(*) = 1
)
) and p.personid in (
  select fp.personid
  from filmparticipation fp natural join film f natural join filmitem i
  where fp.parttype = 'director'
  and i.filmtyp = 'C'
  group by fp.personid
  having max(f.prodyear)-min(f.prodyear) > 49
)
order by name asc;

-- eller
SELECT DISTINCT p.firstname, p.lastname
FROM film f0 NATURAL JOIN filmparticipation fp NATURAL JOIN person p
WHERE fp.parttype = 'director' AND
exists (SELECT f.filmid
        FROM film f NATURAL JOIN filmitem fi NATURAL JOIN
filmparticipation fp1
        WHERE fp1.parttype = 'director' AND fi.filmtyp='C' AND
fp1.personid = fp.personid AND f.filmid
        IN (SELECT f2.filmid
            FROM film f2 NATURAL JOIN filmitem fi2 NATURAL JOIN
filmparticipation fp3
            WHERE fp3.parttype='director' AND fi2.filmtyp='C' AND
f2.filmid
            IN (SELECT f3.filmid
                FROM film f3 NATURAL JOIN filmitem fi3 NATURAL
JOIN filmparticipation fp4
                WHERE fi3.filmtyp = 'C'
                GROUP BY f3.filmid
                HAVING count(distinct fp4.personid) > 150)
```

```

        GROUP BY f2.filmid
        HAVING count(fp3.parttype) = 1))
GROUP BY p.firstname, p.lastname
HAVING (max(f0.prodyear) - min(f0.prodyear) > 49)
ORDER BY p.firstname, p.lastname DESC

-- eller med INTERSECT:
SELECT firstname, lastname
FROM person p NATURAL JOIN filmparticipation fp2 NATURAL JOIN
    (SELECT fp.filmid
     FROM filmparticipation fp, (
         SELECT fp1.filmid
         FROM filmparticipation fp1 NATURAL JOIN filmitem fi1
         WHERE fi1.filmtype = 'C'
         GROUP BY fp1.filmid
         HAVING count(distinct personid) > 150) t1
     WHERE parttype = 'director' AND fp.filmid=t1.filmid
     GROUP BY fp.filmid
     HAVING count(fp.filmid) = 1) t2
WHERE fp2.parttype = 'director'
INTERSECT
SELECT firstname, lastname
FROM person p NATURAL JOIN filmparticipation fp NATURAL JOIN film f NATURAL JOIN
filmitem fi
WHERE fp.parttype = 'director'
AND fi.filmtype = 'C'
GROUP BY p.personid, firstname, lastname
HAVING (max(f.prodyear) - min(f.prodyear) > 49)
ORDER BY firstname, lastname;

```