Øving 2: Relasjonsmodellen, del 2

## Oppgave 1

SELECT tittel, forlag\_id  
FROM bok  
WHERE forlag\_id = 7;

Resultat:  
Et bilde som inneholder tekst, skjermbilde, Font, line

Automatisk generert beskrivelse

### SELECT \* FROM forlag, bok; Resultat: Resultatet ble en lang tabell sammensatt av attributtene fra forlag og bok (forlags attributter til venstre, boks til høyre). Tabellen viste alle kombinasjoner mellom forlag og bok.

1. Likhetsforening (equijoin):  
     
   SELECT b.\*, forlag\_navn, adresse, telefon  
   FROM bok b, forlag f

WHERE b.forlag\_id = f.forlag\_id;

Resultat:

Resultatet forteller at tabellene bok og forlag ble slått sammen (joined) basert på om fremmednøkkelen i bok og primærnøkkelen i forlag er like.

Et bilde som inneholder tekst, skjermbilde, nummer, Font

Automatisk generert beskrivelse  
  
Naturlig forening (Natural join):

SELECT b.\*, forlag\_navn, adresse, telefon   
FROM bok b NATURAL JOIN forlag;

Resultat:  
  
Ved å forsøke å utføre en naturlig forening på bok og forlag ser det ut til at vi får samme resultat som en likhetsforening. Dette skjer fordi med NATURAL JOIN leter automatisk etter samme attributt navn, og matcher verdiene med hverandre.

Et bilde som inneholder tekst, skjermbilde, nummer, Font

Automatisk generert beskrivelse

1. For eksempel er attributtet *tittel* til bok-tabellen unionskompatibel med attributtet *forlag\_navn* til forlag-tabellen. Et annet eksempel kan være attributtet *fornavn* til forfatter-tabellen, og *etternavn* til konsulent-tabellen. Vi kan dermed gjøre kombinasjoner med fornavn- og etternavn attributtene fra både forfatter- og konsulent tabellene unionskompatible. Så lenge attributtene har samme datatype er attributtene unionskompatible.   
     
   Enkelte relasjonsoperasjoner krever at operandene er unionskompatible. Disse operasjonene er nemlig; UNION, INTERSECT (snitt) og DIFFERENCE (differanse).  
     
   Union:  
     
   SELECT tittel AS navn FROM bok  
   UNION  
   SELECT forlag\_navn AS navn FROM forlag;  
     
   Resultat:  
     
   Resultatet fra denne spørringen vil vise en tabell med ett attributt av alle titlene i bok-tabellen og forlag-navnene i forlag-tabellen.  
     
   INTERSECT (Snitt):  
     
   SELECT b.bok\_id, b.tittel  
   FROM bok b  
   WHERE b.bok\_id IN   
   (SELECT forfatter\_id  
   FROM forfatter);  
     
   Resultat:  
     
   Resultatet fra denne spørringen vil vise en tabell over bok id-er og bok-titler for alle bøker med en id som tilhører mengden av forfatter id-ene.  
     
   DIFFERENCE (differanse)  
     
   SELECT b.bok\_id, b.tittel  
   FROM bok b  
   WHERE b.bok\_id NOT IN   
   (SELECT forfatter\_id  
   FROM forfatter);  
     
   Resultatet fra denne spørringen vil vise en tabell over bok id-er og bok-titler for alle bøker der hvor en bok ikke har en id i mengden av forfatter id-er.

## Oppgave 2

1. SELECT fornavn, etternavn  
   FROM forfatter  
   ORDER BY etternavn  
     
   For å finne fornavn og etternavn til alle forfatterne sortert alfabetisk brukte jeg ORDER BY klausulen for å sortere listen etter etternavn.
2. SELECT f.forlag\_navn, f.forlag\_id  
   FROM forlag f   
   WHERE f.forlag\_id NOT IN   
   (SELECT b.forlag\_id FROM bok b);  
     
   Et bilde som inneholder tekst, skjermbilde, nummer, Font

   Automatisk generert beskrivelse  
   For å finne forlag som ikke har utgitt bøker. Brukte jeg operasjonen SUB-SELECT for å finne forlag id-er som ikke finnes i bøker.
3. SELECT f.forfatter\_id, f.fornavn, f.etternavn, f.fode\_aar  
   FROM forfatter f  
   WHERE f.fode\_aar < 1900  
   ORDER BY f.forfatter\_id  
   Et bilde som inneholder tekst, Font, nummer, line

   Automatisk generert beskrivelse  
   For å finne forfattere som er født før 1900 brukte jeg klausulen WHERE sammen med sammenliknings operatoren <. I tillegg valgte jeg å sortere tuplene etter forfatter\_id siden jeg lå merke til at det var flere tupler med samme forfatter fornavn og etternavn.
4. SELECT DISTINCT f.forlag\_navn, f.adresse  
   FROM forlag f, bok b  
   WHERE b.tittel = 'sult' AND b.forlag\_id = f.forlag\_id  
     
   Et bilde som inneholder tekst, skjermbilde, Font, line

   Automatisk generert beskrivelse  
   For å finne navn og adresse til forlaget som har utgitt boka Sult, brukte jeg WHERE klausulen til og den logiske operatoren AND for å finne boktittel og forlags id som matcher. I tillegg bruker DISTINCT slik at jeg unngår å få duplikater i tabellen.
5. SELECT DISTINCT b.tittel  
   FROM bok b, forfatter f, bok\_forfatter bf

WHERE bf.bok\_id = b.bok\_id AND bf.forfatter\_id IN (

SELECT f.forfatter\_id FROM forfatter f   
WHERE f.etternavn = 'Hamsund');

Et bilde som inneholder tekst, skjermbilde, Font, nummer

Automatisk generert beskrivelse

For å finne titlene på bøkene som Hamsund har skrevet, brukte jeg WHERE klausulen sammen med AND for å sammenlikne bok id-er og forfatter id-er i tabellen bok\_forfatter. Jeg brukte også operasjonen SUB-SELECT for å finne de id-ene som var knyttet til etternavnet Hamsund.

1. SELECT b.bok\_id, b.tittel, b.utgitt\_aar, f.\*   
   FROM bok b RIGHT JOIN forlag f   
   ON b.forlag\_id = f.forlag\_id  
   Et bilde som inneholder tekst, skjermbilde, nummer, meny

   Automatisk generert beskrivelse

For å gi full oversikt over bøker og forlag som har utgitt dem, brukte jeg operasjonen RIGHT JOIN. Grunnen til at jeg brukte RIGHT JOIN er fordi da vil forlag som ikke har utgitt vises med NULL verdier for bøker.