

# Øving 3 - SQL del 1

## Oppgave 1

Sett opp `SELECT`-setninger som besvarer spørsmålene nedenfor. Kun én setning pr oppgave.

### 1.01

1. Finn alle borettslag etablert i årene 1975-1985.

Velger `SELECT` kolonnene 'bolag\_navn' og `etabl_aar` fra tabellen `borettslag`.

Setter filteret `WHERE` på verdien `etabl_aar`, og spesifiserer at det skal være verdier mellom 1975 og 1985.

```
1 SELECT bolag_navn, etabl_aar
2 FROM borettslag
3 WHERE etabl_aar
4 BETWEEN 1975 AND 1985;
```

Dette gir ut følgende tabell:

bolag_navn	etabl_aar
Tertitten	1980

### 1.02

2. Skriv ut en liste over andelseiere. Listen skal ha linjer som ser slik ut (tekster i kursiv er data fra databasen):

*"fornavn etternavn, ansiennitet: ansiennitet år"*.

Listen skal være sortert på ansiennitet, de med lengst ansiennitet øverst.

Bruker i stor grad en vanlig `SELECT`-setning, men bygger også en streng via `CONCAT` ("concatenation").

Videre brukes `ORDER BY` til sortering, og `DESC` for å sortere synkende ("descending").

```
1 SELECT CONCAT(fornavn, ' ', etternavn, ', ansiennitet: ', ansiennitet, ' år')
2 FROM andelseier
3 ORDER BY ansiennitet DESC;
```

Dette gir følgende output:

```

1 CONCAT(fornavn, ' ', etternavn, ', ansiennitet: ', ansiennitet, ' år')
2
3 Anna Olsen, ansiennitet: 10 år
4 Ingrid Olsen, ansiennitet: 8 år
5 Arne Torp, ansiennitet: 7 år
6 Arne Martinsen, ansiennitet: 4 år
7 Even Trulsbo, ansiennitet: 3 år

```

## 1.03

3. I hvilket år ble det eldste borettslaget etablert?

Velger kolonnene `bolag_navn` og `etabl_aar` fra tabellen `borettslag`. Sorterer deretter på `etabl_aar`, og velger stigende, siden vi da vil få den eldste først, altså det laveste tallet.

```

1 SELECT bolag_navn, etabl_aar
2 FROM borettslag
3 ORDER BY etabl_aar ASC;

```

Dette gir følgende output:

<code>bolag_navn</code>	<code>etabl_aar</code>
Tertitten	1980
Sisiken	1990
Lerken	2000

Fra dette ser vi fort at det eldste borettslaget, Tertitten, ble etablert i 1980.

Kanskje en "for enkel" måte å gjøre det på, men det fungerer.

## 1.04

4. Finn adressene til alle bygninger som inneholder leiligheter med minst tre rom.

Bruker `JOIN` for å flette sammen kolonner fra tabellene `leilighet` og `borettslag` på kolonnen `bygn_id`. Deretter filtreres den nye tabellen på `leilighet.ant_rom` større enn eller lik 3 (`>= 3`, altså minst 3.)

```

1 SELECT DISTINCT bygning.bygn_adr
2 FROM bygning
3 JOIN leilighet ON bygning.bygn_id = leilighet.bygn_id
4 WHERE leilighet.ant_rom >= 3;

```

Resultatet er følgende tabell:

ant_rom	bygn_id	bolag_navn
5	<a href="#">1</a>	Lerken
5	<a href="#">1</a>	Tertitten
5	<a href="#">1</a>	Sisiken
5	<a href="#">1</a>	Lerken
5	<a href="#">1</a>	Tertitten
5	<a href="#">1</a>	Sisiken
5	<a href="#">1</a>	Lerken
5	<a href="#">1</a>	Tertitten
5	<a href="#">1</a>	Sisiken

Riktig:

Åsveien 100a

## 1.05

5. Finn antall bygninger i borettslaget "Tertitten".

### Løsning 1: Manuell telling

asd

```

1 SELECT bygn_id
2 FROM bygning
3 JOIN borettslag ON bygn_id
4 WHERE borettslag.bolag_navn LIKE "Tertitten";

```

Denne løsningen skriver ut en liste med alle bygnings-ID som tilhører oppføringene med `bolag_navn` "Tertitten".

bygn_id
1
2
3
4
5

Vi kan da telle oppføringene, eller bare se på det høyeste nummeret av ID-nummer. En "enkel" løsning, som fungerer.

## Løsning 2: Bruke COUNT

Det er også mulig å skrive direkte ut hvor mange linjer/rader det er i tabellen ved å bruke `COUNT`:

```
1 SELECT COUNT(*) as bygn_id
2 FROM bygning
3
4 WHERE bygning.bolag_navn LIKE "Tertitten";
```

Dette gir output:

bygn_id
5

Altså får vi ut hvor mange rader det er i tabellen som ble konstruert ut fra seleksjon of filter.

## 1.06

6. Lag en liste som viser antall bygninger i hvert enkelt borettslag. Listen skal være sortert på borettslagsnavn. Husk at det kan finnes borettslag uten bygninger - de skal også med.

Bruker `SELECT` for å velge kolonnene `borettslag.bolag_navn` og `bygning.bygn_id`. Bruker `COUNT` for å telle opp antall oppføringer i listen `bygning.bygn_id` og grupperer dem med `GROUP BY` etter `bolag_navn` slik at vi får hvert antall tilhørende hvert borettslag basert på navn. Setter nytt navn på kolonnen med `AS antall_bygninger`.

`LEFT JOIN` sørger for å få med borettslag uten bygninger. Siden det ikke er noen oppføringer på `bygn_id` i et borettslag uten bygninger, blir antallet injert der 0.

```
1 SELECT
2     borettslag.bolag_navn,
3     COUNT(bygning.bygn_id) AS antall_bygninger
4 FROM borettslag
5 LEFT JOIN bygning ON borettslag.bolag_navn = bygning.bolag_navn
6 GROUP BY borettslag.bolag_navn
7 ORDER BY borettslag.bolag_navn;
```

Dermed kommer følgende tabell ut:

bolag_navn	antall_bygninger
Lerken	0

bolag_navn	antall_bygninger
Sisiken	1
Tertitten	4

## 1.07

7. Finn antall leiligheter i borettslaget "Tertitten".

Forsøk #1 med sine mangler

```

1 SELECT
2     bygning.bygn_id,
3     COUNT(leilighet.leil_nr) AS antall_leiligheter,
4 FROM borettslag
5     JOIN bygning ON borettslag.bolag_navn = bygning.bolag_navn
6 WHERE borettslag.bolag_navn = 'Tertitten'
7 GROUP BY borettslag.bygn_id
8 ORDER BY borettslag.bolag_navn;
```

Etter prøving og feiling ble denne fungerende:

```

1 SELECT
2     bygning.bygn_id,
3     COUNT(leilighet.leil_nr) AS antall_leiligheter
4 FROM borettslag
5 JOIN bygning ON borettslag.bolag_navn = bygning.bolag_navn
6 JOIN leilighet ON bygning.bygn_id = leilighet.bygn_id
7 WHERE borettslag.bolag_navn = 'Tertitten'
8 GROUP BY bygning.bygn_id
9 ORDER BY borettslag.bolag_navn;
```

Og med kolonnen `bolag_navn`, for å gjøre det enklere å lese:

```

1 SELECT
2     borettslag.bolag_navn,
3     bygning.bygn_id,
4     COUNT(leilighet.leil_nr) AS antall_leiligheter
5 FROM borettslag
6 JOIN bygning ON borettslag.bolag_navn = bygning.bolag_navn
7 JOIN leilighet ON bygning.bygn_id = leilighet.bygn_id
8 WHERE borettslag.bolag_navn = 'Tertitten'
9 GROUP BY borettslag.bolag_navn, bygning.bygn_id
10 ORDER BY borettslag.bolag_navn;
```

Denne spørringen gir følgende tabell:

bolag_navn	bygn_id	antall_leiligheter
Tertitten	1	3
Tertitten	3	1

Vi ser her at bygning 1 og 3 har leiligheter. Bygning 1 har 3 leiligheter, og bygning 3 har 1 leilighet. Kan verifisere dette ved å sjekke tabellen for leiligheter manuelt.

leil_nr	ant_rom	ant_kvm	etasje	bygn_id	and_eier_nr
1	5	110	3	1	1
2	5	110	3	1	2
3	2	50	1	3	3
4	5	110	1	1	4

## 1.08

8. Hvor høyt kan du bo i borettslaget "Tertitten"?

Her må vi altså finne den bygningen i Tertitten borettslag med flest etasjer. Her brukes i stor grad samme spørring som i forrige oppgave.

```

1  SELECT
2      borettslag.bolag_navn,
3      bygning.bygn_id,
4      MAX(bygning.ant_etasjer) AS etasjer
5  FROM borettslag
6      JOIN bygning ON borettslag.bolag_navn = bygning.bolag_navn
7      JOIN leilighet ON bygning.bygn_id = leilighet.bygn_id
8  WHERE borettslag.bolag_navn = 'Tertitten'
9  GROUP BY borettslag.bolag_navn, bygning.bygn_id
10 ORDER BY etasjer DESC
11 LIMIT 1;
```

Bruker MAX for å finne den høyeste verdien i tabell.kolonne `bygning.ant_etasjer`.

Bruker `LIMIT 1` i kombinasjon med `ORDER BY etasjer DESC` for å vise kun den bygningen med flest etasjer. Dette fordi synkende sortering vil sette den høyeste verdien øverst, og så begrenser vi tabellen til å vise kun rad nummer `1`.

bolag_navn	bygn_id	etasjer
Tertitten	3	6

Det høyeste man kan bo i Tertitten borettslag er altså i 6. etasje.

## 1.09

9. Finn navn og nummer til andelseiere som ikke har leilighet.

Versjon #1: Finne alle andelseiere og leilighetsnumre, og bare sjekke tabellen for hvilken andelseier som har `NULL` leilighet:

```
1 SELECT
2     andelseier.and_eier_nr, andelseier.fornavn, andelseier.etternavn,
3     leilighet.leil_nr
4 FROM andelseier
5 LEFT JOIN leilighet on andelseier.and_eier_nr = leilighet.and_eier_nr
6 ORDER BY andelseier.and_eier_nr;
```

Dette gir oss alle leietakerene og deres leiligheter. Leser da av at andelseier nr. 5 har `NULL` på leilighetsnummer.

and_eier_nr	fornavn	etternavn	leil_nr
1	Even	Trulsbo	1
2	Anna	Olsen	2
3	Ingrid	Olsen	3
4	Arne	Torp	4
<b>5</b>	<b>Arne</b>	<b>Martinsen</b>	<b>null</b>

For å vise **bare** den som ikke har en leilighet, altså har `NULL` i kolonnen `leil_nr` kan vi bruke filer `IS NULL`:

```
1 SELECT
2     andelseier.and_eier_nr, andelseier.fornavn, andelseier.etternavn,
3     leilighet.leil_nr
4 FROM andelseier
5 LEFT JOIN leilighet on andelseier.and_eier_nr = leilighet.and_eier_nr
6 WHERE leilighet.leil_nr IS NULL
7 ORDER BY andelseier.and_eier_nr;
```

Får da ut følgende tabell:

and_eier_nr	fornavn	etternavn	leil_nr
5	Arne	Martinsen	null

Som i tabellen over, ser vi at andelseier nr. 5 er den som er uten leilighet.

## 1.10

10. Finn antall andelseiere pr borettslag, sortert etter antallet. Husk at det kan finnes borettslag uten andelseiere - de skal også med.

For å finne antallet andelseiere per borettslag, også de borettslagene uten andelseiere, kan `LEFT JOIN` brukes mellom tabellene `borettslag` og `andelseiere`.

```
1 SELECT
2     borettslag.bolag_navn,
3     COUNT(andelseier.and_eier_nr) as antall_andelseiere
4 FROM
5     borettslag
6 LEFT JOIN
7     andelseier ON borettslag.bolag_navn = andelseier.bolag_navn
8 GROUP BY
9     borettslag.bolag_navn
10 ORDER BY
11     antall_andelseiere;
```

Henter data fra tabellen `borettslag` med `FROM`. Videre velger vi `borettslag.bolag_navn`, og teller opp antall andelseiere fra tabell.kolonne `andelseier.and_eier_nr`. Gir denne kolonnen et alias, `antall_andelseiere`.

Bruker deretter `LEFT JOIN` mellom `andelseier` og `borettslag`, og matcher sammenføyningen på `tabell.bolag_navn` for å hente ut andelseiere tilknyttet hvert borettslag.

Bruker `GROUP BY` for å gruppere resultatet etter borettslagnavn, og `ORDER BY` antall andelseiere for å sortere resultatet stigende etter antallet.

Resultat:

bolag_navn	antall_andelseiere
Lerken	0
Sisiken	1
Tertitten	4

## 1.11

11. Skriv ut en liste over alle andelseiere. For de som har leilighet, skal leilighetsnummeret skrives ut.

For å lage en liste som inkluderer alle andelseiere, og leiligheten deres om de har en, kan man benytte `LEFT JOIN` på `andelseier` og `leilighet`.



```

1 SELECT
2     andelseier.and_eier_nr,
3     andelseier.fornavn,
4     andelseier.etternavn,
5     leilighet.leil_nr AS leilighetsnummer
6 FROM
7     andelseier
8 LEFT JOIN
9     leilighet ON andelseier.and_eier_nr = leilighet.and_eier_nr;

```

Gjør en `LEFT JOIN` mellom `andelseier` og `leilighet`, basert på `and_eier_nr` for at alle andelseiere skal bli tatt med i resultatet, også de som ikke har leilighet, altså dem som har `NULL` i kolonnen `leilighetsnummer`.

Velger `and_eier_nr`, `fornavn` og `etternavn` fra tabellen `andelseier`, og `leil_nr` fra tabellen `leilighet`.

Vi får dermed ut denne tabellen:

and_eier_nr	fornavn	etternavn	leilighetsnummer
1	Even	Trulsbo	1
2	Anna	Olsen	2
3	Ingrid	Olsen	3
4	Arne	Torp	4
5	Arne	Martinsen	null

## 1.12

12. Hvilke borettslag har leiligheter med eksakt 4 rom?

Å finne hvilke borettslag som har leiligheter med eksakt 4 rom kan gjøres med følgende spørring:

```

1 SELECT
2     borettslag.bolag_navn,
3     leilighet.ant_rom,
4     leilighet.leil_nr
5 FROM
6     borettslag
7     INNER JOIN
8     bygning ON borettslag.bolag_navn = bygning.bolag_navn
9     INNER JOIN
10    leilighet ON bygning.bygn_id = leilighet.bygn_id
11 WHERE
12     leilighet.ant_rom = 4
13 GROUP BY

```

```
14     borettslag.bolag_navn,  
15     leilighet.ant_rom;
```

Resultatet her er en tom tabell, altså ingen borettslag har leiligheter med eksakt 4 rom.

For å sjekke litt mer kan man endre `WHERE` på linje 12 til å være `leilighet.ant_rom > 0`, for å vise alle leiligheter som har flere enn 0 rom. Vi ser da at ingen har 4 rom:

bolag_navn	ant_rom	leil_nr
Tertitten	2	3
Tertitten	5	1

## 1.13

13. Skriv ut en liste over antall andelseiere pr postnr og poststed, begrenset til de som bor i leiligheter tilknyttet et borettslag. Husk at postnummeret til disse er postnummeret til bygningen de bor i, og ikke postnummeret til borettslaget. Du trenger ikke ta med poststeder med 0 andelseiere. (Ekstraoppgave: Hva hvis vi vil ha med poststeder med 0 andelseiere?)

Spørring for liste over antall andelseiere per postnummer og poststed, begrenset til de som bor i leiligheter tilknyttet et borettslag. Inkluderer også navn på borettslaget, siden oppgaven sier de skal være tilknyttet et borettslag.

```
1  SELECT  
2      poststed.postnr,  
3      poststed.poststed,  
4      borettslag.bolag_navn,  
5      COUNT(andelseier.and_eier_nr) as antall_andelseiere  
6  FROM  
7      andelseier  
8  INNER JOIN  
9      leilighet ON andelseier.and_eier_nr = leilighet.and_eier_nr  
10 INNER JOIN  
11     bygning ON leilighet.bygn_id = bygning.bygn_id  
12 INNER JOIN  
13     poststed ON bygning.postnr = poststed.postnr  
14 INNER JOIN  
15     borettslag ON bygning.bolag_navn = borettslag.bolag_navn  
16 GROUP BY  
17     poststed.postnr, poststed.poststed, borettslag.bolag_navn  
18 HAVING  
19     COUNT(andelseier.and_eier_nr) > 0;
```

Velger hovedtabellen med `FROM`, og velger kolonnene vi vil se på med `SELECT`. Teller opp antall andelseiere med `COUNT`.

Utfører flere `INNER JOIN`-operasjoner, som tar snittet/fellesverdier av kolonner, for å slå sammen kolonner fra tabellen `andelseier` med kolonner fra de oppgitte tabellene, og matche dem på nøklene som blir gitt i `ON tabellA.kolonne_ID = tabellB.kolonne_ID`-setningen.

Grupperer og fjerner de oppføringene som har 0 andelseiere.

Resultat:

postnr	poststed	bolag_navn	antall_andelseiere
7020	Trondheim	Tertitten	4

Ekstraoppgave:

Bruker `LEFT JOIN` i stedet, slik at alt som ikke har tilknyttede borettslag eller andelseiere blir med.

```
1  SELECT
2      poststed.postnr,
3      poststed.poststed,
4      borettslag.bolag_navn,
5      COUNT(andelseier.and_eier_nr) as antall_andelseiere
6  FROM
7      poststed
8  LEFT JOIN
9      bygning ON poststed.postnr = bygning.postnr
10 LEFT JOIN
11     borettslag ON bygning.bolag_navn = borettslag.bolag_navn
12 LEFT JOIN
13     leilighet ON bygning.bygn_id = leilighet.bygn_id
14 LEFT JOIN
15     andelseier ON leilighet.and_eier_nr = andelseier.and_eier_nr
16 GROUP BY
17     poststed.postnr, poststed.poststed, borettslag.bolag_navn;
```

Resultat:

postnr	poststed	bolag_navn	antall_andelseiere
2020	Skedsmokorset	null	0
6408	Aureosen	null	0
7020	Trondheim	Sisiken	0
7020	Trondheim	Tertitten	4
7033	Trondheim	null	0