

## Øving 3: Normalisering og SQL del 1 (obligatorisk)

Tore Mallaug, NTNU  
Innleveringsfrist: se Blackboard  
Tidligste godkjenning: datoer blir annonsert

Løsningsforslag legges ut i etterkant.

Alle obligatoriske øvinger må være godkjente for å få karakter i emnet.

### Oppgave 1 Normalisering

Et firma som leier ut feriehus lagrer følgende informasjon knyttet til utleie:

kundedata: navn, adresse, telefon

data om eiendommen: adresse

data om eieren: navn, adresse, telefon

data om utleieforholdet: fra og med uke, til og med uke, pris pr uke

Gitt at vi har én tabell med 13 attributter (kolonner) som ser altså slik ut (skrevet på relasjonell form uten nøkler):

```
leieforhold (kunde_id, kunde_navn, kunde_adresse, kunde_tlf,
eiendom_id, eiendom_adresse, eier_id, eier_navn, eier_adresse,
eier_tlf, fra_uke, til_uke, pris)
```

I tillegg til dataene nevnt innledningsvis er det lagt inn kunde\_id, eier\_id og en eiendoms\_id, som entydig identifiserer henholdsvis en person og en eiendom.

Foreslå kandidatnøkler for denne tabellen. Anta at en person kun kan leie en eiendom av gangen, og at en eiendom kan leies ut til kun en person av gangen.

Tabellen er ikke problemfri mht til registrering og sletting av data. Forklar hvorfor.

Tegn ett (kun ett) diagram (tilsvarende figur som i læreboka kap. 3) som viser funksjonelle avhengigheter mellom alle attributtene.

Du skal nå bruke funksjonelle avhengigheter og BCNF til å foreslå en oppsplitting av tabellen i mindre tabeller slik at problemene vedr. registrering og sletting av data unngås.

Sett opp, direkte fra figuren, relasjoner som er på BCNF.

Kan vi løse denne oppgaven ved å gjennomføre prosessen 1NF --> 2NF --> 3NF? Begrunn svaret.

## Oppgave 2 SQL

### Database

I denne oppgaven skal vi bruke en borettslag-databasen. Bruk følgende sql-script: [MySQL](#). Scriptet inneholder data som oppgavene nedenfor spør etter.

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

1. Finn alle borettslag etablert i årene 1975-1985.
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.
3. I hvilket år ble det eldste borettslaget etablert?
4. Finn adressene til alle bygninger som inneholder leiligheter med minst tre rom.
5. Finn antall bygninger i borettslaget "Tertitten".
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.
7. Finn antall leiligheter i borettslaget "Tertitten".
8. Hvor høyt kan du bo i borettslaget "Tertitten"?
9. Finn navn og nummer til andelseiere som ikke har leilighet.
10. Finn antall andelseiere pr borettslag, sortert etter antallet. Husk at det kan finnes borettslag uten andelseiere - de skal også med.
11. Skriv ut en liste over alle andelseiere. For de som har leilighet, skal leilighetsnummeret skrives ut.
12. Hvilke borettslag har leiligheter med eksakt 4 rom?
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?)