

# Øving 1

## Oppgave 1: Databasesystemer

- a) En database er en samling med data, gjerne organisert som kolonner med informasjon i rader. Et databasehåndteringssystem (DBMS) er et verktøy brukt for å lage og 'drifte' databaser. Her lagres ofte datas format slik at den enkelt kan endres og leses.
- b)
- 1) Program-data uavhengighet: Fører til at data kan endres uten at det vil kunne tvinge selve applikasjonene som bruker den til å måtte endres i tillegg; fordi måten dataen aksesseres på er skrevet uavhengig av programmet.
  - 2) Flerbrukerstøtte: Gir muligheten til at flere brukere kan endre databasen samtidig. Noe som er en essensiell egenskap for å få nytten av å bruke en database. Naturligvis er det inkludert funksjoner slik at
  - 3) Selvbeskrivende: I enkelte databaser unnlater man å beskrive metadata i DBMSen, her er dataen selvbeskrivende slik at det ikke trenges en egen katalog for metadataen.

## Oppgave 2: ER-modellen

- a)
- 1) En entitet er et objekt i databasen, altså noe som eksisterer der. En entitetsklasse derimot er en samling av disse entitetene med like attributter og egenskaper.
  - 2) En relasjon er en sammenheng mellom ulike/flere entiteter. En relasjonsklasse er en mengde av like relasjoner mellom de samme entitetsklassene.
  - 3) Alle entiteter må ha en nøkkelattributt så de kan skilles fra hverandre. Denne fungerer som en unik identifikator for hver enkelt entitet.

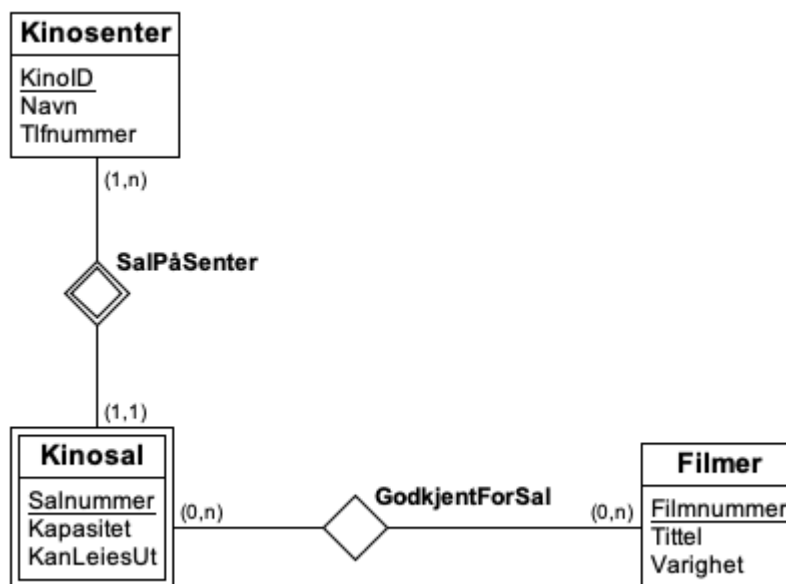
b)

Påstand	Svar	Begrunnelse
1. Taco er en entitetsklasse og hver enkelt instans av en taco identifiseres ved hjelp av nøkkelattributtet TacoID.	True	Entitetsklassen er definert med TacoID som en nøkkelattributt, denne brukes for å identifisere enkelte entiteter.
2. En taco kan inneholde et vilkårlig antall kjøttdeiger, oster, grønnsaker og sauser.	True	Alle disse attributtene har en null-til-mange relasjon med Taco.
3. En ordre gjort av en kunde trenger ikke inneholde noen taco.	False	Denne relasjonsklassen er definert som en én-til-mange relasjon, dermed må en ordre inneholde minst én taco.
4. En ordre gjort av en kunde kan inneholde en million tacoer.	True	Det er ingen begrensninger på antall relasjoner til ordren.
5. Så fort en ordre er opprettet kan den hentes i en vilkårlig butikk.	False	Ordre entitetsklassen har én-til-én forhold til butikk.
6. En kunde kan eksistere i systemet uten noensinne å ha bestilt noen tacoer.	True	En kunde være lagret i databasen som en entitet og trenger ingen relasjoner til seg.
7. Hver eneste grønnsak-entitet har en vekt.	True	Relasjonsklassen til 'TacoGrønnsak' beskrives med vekt.
8. En ansatt kan jobbe i ulike butikker med ulike stillingstitler.	True	Denne relasjonsklassen er definert som en én-til-mange relasjon.
9. Alle ordre delegeres typisk til denne samme personen som er ansatt.	Maybe	Ordrer delegeres til én person, men vi har ikke informasjon om hvilke ansatt vil få ordren.
10. En kunde må være registrert med et navn.	True	Alle attributter må utfylles.

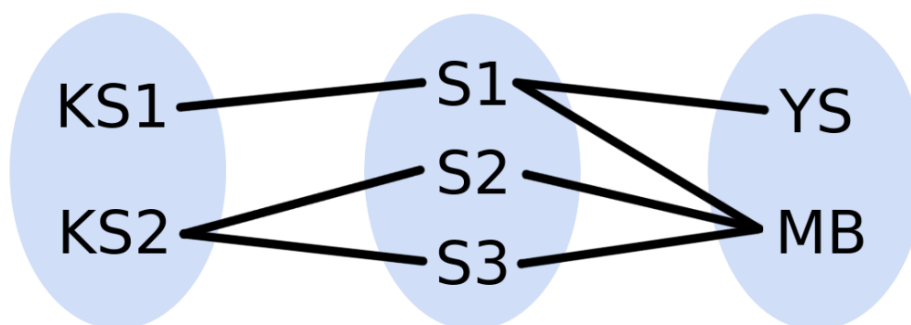
### Oppgave 3: Svake klasser, forekomstdiagram og nye krav

- a) Den identifiserende entitetsklassen er “Kinosenter”, den identifiserende relasjonsklassen er “SalPåSenter” og delvis nøkkel er “Salnummer”. Det kan være hensiktsmessig å bruke svake entitetsklasser når en nøkkelattributt ikke er nødvendig; hvor entiteten heller kan identifiseres gjennom dets relasjoner.
- b) Dersom vi endrer kardinalitetet til å være null-til-én vil ikke salen kunne identifiseres ettersom den ikke lenger *må* høre til et kinosenter. Flere kinosenter kan heller ikke ha den samme salen som ville bli muliggjort med en én-til-mange relasjon.

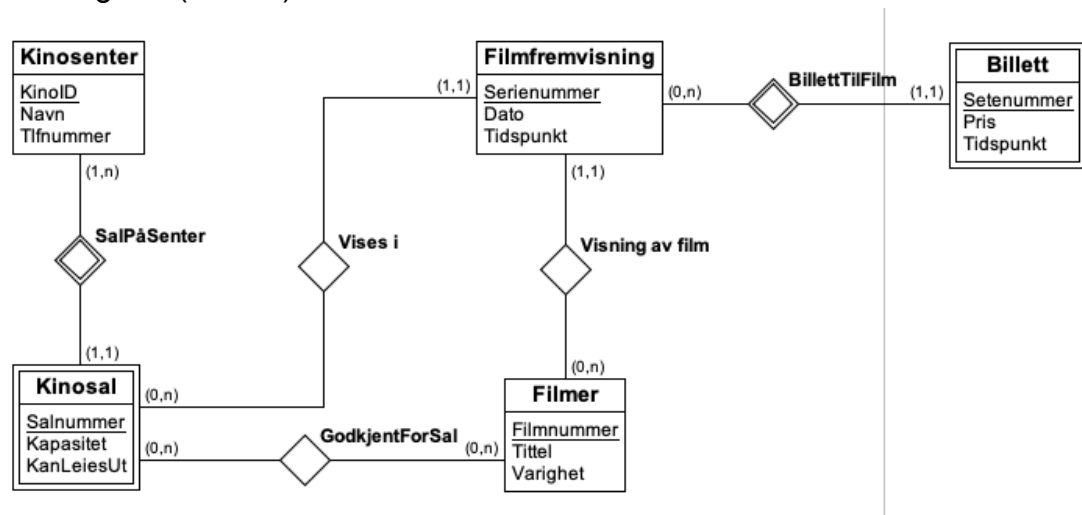
c)



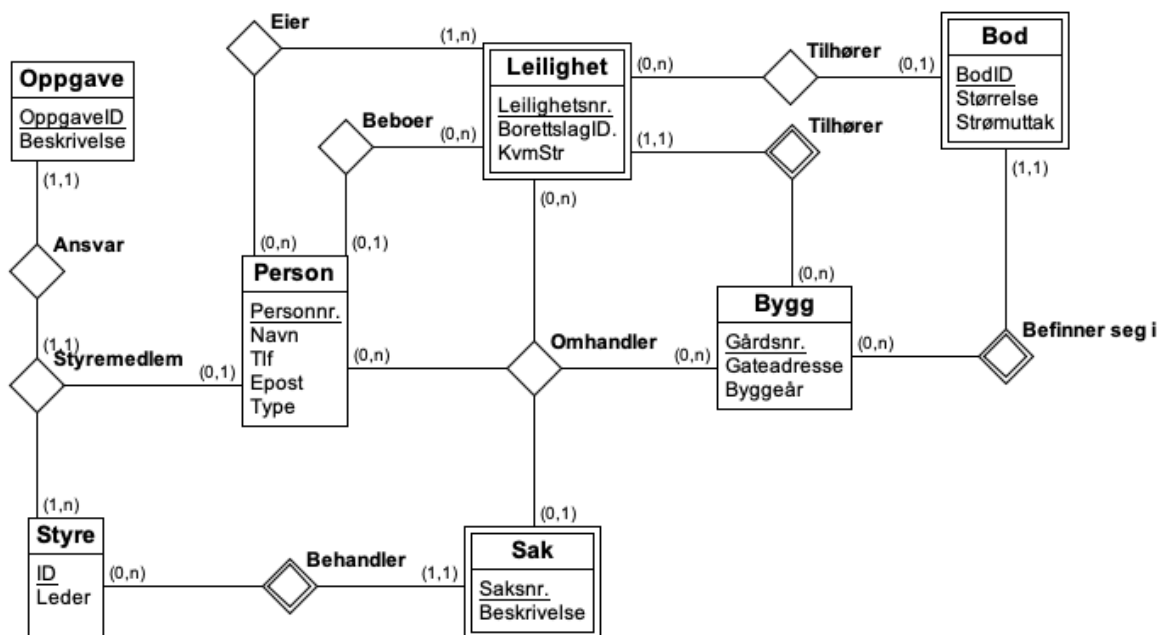
d)



### e) ER-Diagram (utvidet)



### Oppgave 4: Fra miniverdenen til ER-modell



Antagelser:

- En bod må ikke tilhøre en leilighet (eks. fellesbod)
- En person må ikke bo i borettslaget for å kunne eie en leilighet (eks. foreldre som eier for sine barn)