

TDT4145 Øving 1

Philip Ditlevsen, Yngve Tryggestad Larsen, Natalie Sørensen Forshaw

Januar 2021

1 Databasesystemer

a) En database er en strukturert og systematisk samling av data, ofte tilknyttet et elektronisk medium.

Databasehåndteringssystem er programvare som er designet for å håndtere databasene. Den skal kunne håndtere lagring, søking, henting og oppdatering av dataen i databasen.

b) Program-data uavhengighet: Når det må gjøres endringer i en database er det program-data uavhengighet-egenskapen som sørger for at dataen fortsatt er intakt og tilgjengelig. Dette tillater en eventuell bedrift å gjøre endringer på et av nivåene i en database uten at det andre nivået påvirkes, dermed kan brukeren av databasen forholde seg upåvirket mens endringen gjennomføres. Det er to typer av disse nivåene, fysisk og logisk. Fysisk data uavhengighet referer til når det foregår en eventuell maskinvare oppgradering uten å forstyrre for det logiske nivået i databasen, så brukerne har fortsatt tilgang på all dataen gjennom sin datamaskin. Logisk data uavhengighet gir bedriften muligheten til å oppgradere det logiske nivået, som å legge til mer data eller flere tabeller uten å gå utover det fysiske nivået. Fordelen med denne egenskapen er at vedlikehold av databasen er mye enklere, det hemmer ikke brukerne når endringer må gjøres, og det krever mindre arbeid å utføre siden en endring på et nivå ikke behøver justeringer på det andre nivå.

Flerbrukerstøtte: En database med flerbrukerstøtte kan benyttes av flere brukere samtidig. Fordelen er fortgang fordi man ikke behøver å vente på tur, operasjoner fra flere brukere kan kjøre parallelt.

Selvbeskrivende: En selvbeskrivende database inneholder ikke bare dataen, men også en beskrivelse av dataen og strukturen. Disse beskrivelsene er metadata, og kan beskrive for eksempel dataens filtype eller størrelse. Fordelen er at informasjonen er tilgjengelig om man skulle få bruk for den, fremfor at man ikke vet noe om dataen i databasen.

2 ER-modellen

a)

1) Forskjellen på en entitet og en entitetsklasse er at en entitet er et objekt i en miniverden, mens en entitetsklasse er mengden av likartede entiteter som deler type og egenskaper.

2) Forskjellen på en relasjon og en relasjonsklasse er at en relasjon mellom 2 eller flere entiteter mens en relasjonsklasse er mengden av likartede relasjoner mellom samme entitetsklasser.

3) Alle entiteter må ha en eller flere nøkkelattributter fordi nøkkelattributter er entydige identifikatorer, det fungerer som en id for entiteten. Om vi vil aksisere en entitet med en spesifikk nøkkelattributt så vet vi at det er nøyaktig bare denne ene entiteten som har denne attributten.

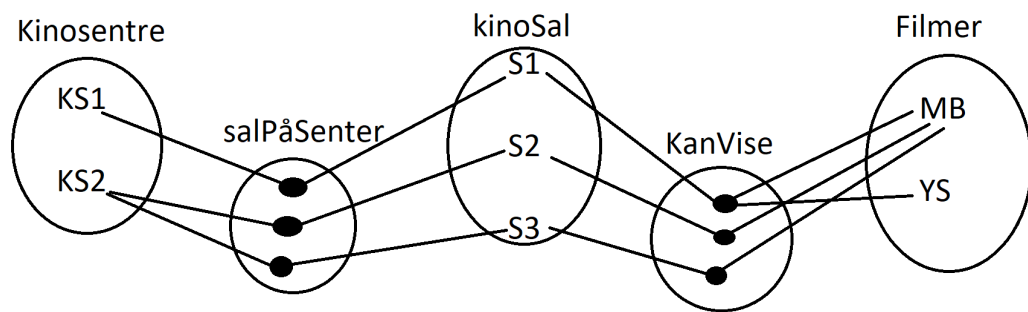
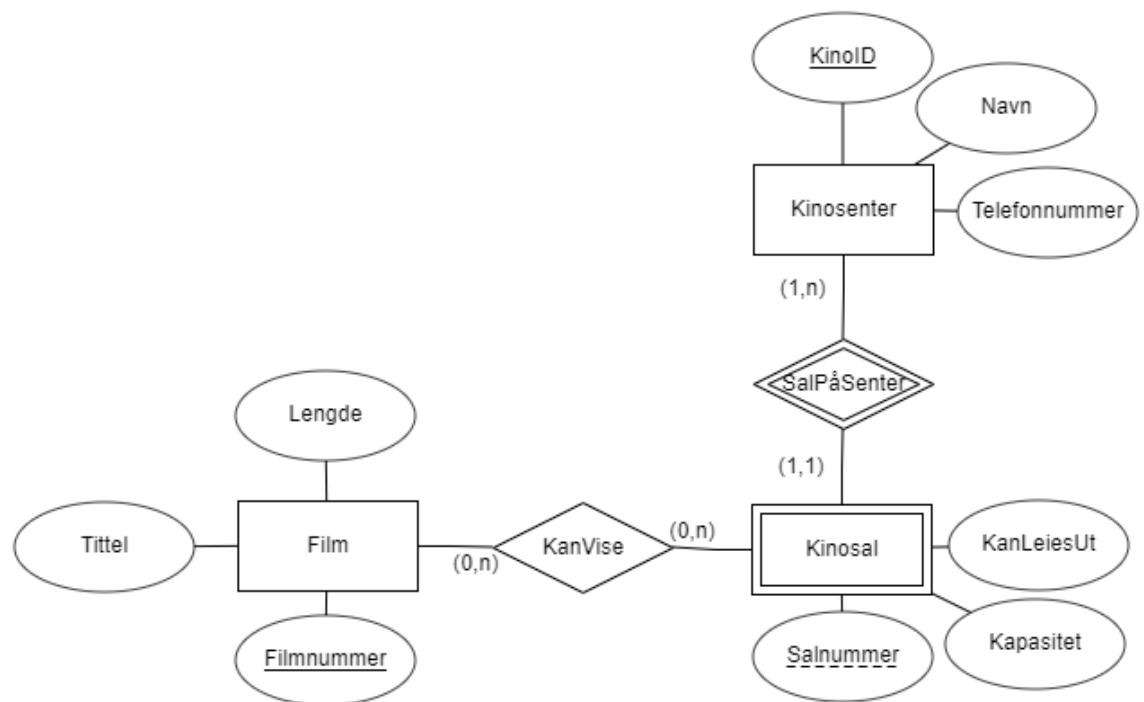
Påstand nr.	Svar	Begrunnelse
1	True	Taco er en entitetsklasse som kan ha mange ulike taco entiteter, og TacoID er unik for alle tacoer, altså en nøkkelattributt.
2	True	Kardinaliteten mellom taco og kjøttdeiger, oster, grønnsaker og sauser er (0,n). Så tacoen kan inneholde et vilkårlig antall av disse ingrediensene.
3	False	Kardinaliteten mellom ordre og en taco er (1,n) altså det er minst en taco per ordre
4	True	Kardinaliteten viser at en ordre kan inneholde (1,n) så av den grunn kan en ordre inneholde 1 million tacoer
5	False	Kardinaliteten til en ordre og en butikk er (1,1). Så det kan kun være 1 butikk per ordre
6	True	Kardinaliteten mellom kunden og ordre er (0,n), så det er ikke et krav at en kunde har en ordre
7	False	Vekten til grønnsaken blir deklartert i relasjonen mellom taco og grønnsak. Vekten er knyttet til relasjonen TacoGrønnsak ikke entiteten, derfor vil ikke vekten eksistere om relasjonen ikke eksisterer
8	True	Ettersom "Jobber"-relasjonen har attributten stillingstittel, vil en ansatt kunne ha relasjon til ulike arbeidsplasser hvor stillingstittlen er unik for hver relasjon
9	Maybe	Dette er noe som vi ikke kan vite noe om basert på informasjonen i diagrammet. Ingenting i diagrammet indikerer at det fungerer slik, da det er opp til logikken utenfor databasen å håndtere delegeringen av ordre.
10	Maybe	I utgangspunktet kan en bruker uten navn eksistere, men logikken utenfor databasen kan gjøre at en konto uten navn ikke tillates.

3 Svake klasser, forekomstdiagram og nye krav

a) Svake entitetsklasser er logiske å bruke når man vet at en entitet er avhengig av en annen entitet for å eksistere. Som i eksempelet, kan ikke en kinosal eksistere uten et kinosenter. Det vil også være fint å bruke hvor det er fordelaktig at en entitet arver primærnøkkel fra en annen. Det er nyttig dersom en attributt ikke er en entydig identifikator, men i relasjon med en identifiserende entitetsklasse kan den være unik innenfor denne mini-verdenen. I eksempelet er identifiserende entitetsklasse Kinosenter, og identifiserende relasjonsklasse er SalPåSenter. Det er salnummer som er den delvise nøkkelen. Salnummeret er ikke unikt, det eksisterer på flere kinosentre, men innenfor et spesifikt kinosenter er det den eneste salen med dette salnummeret.

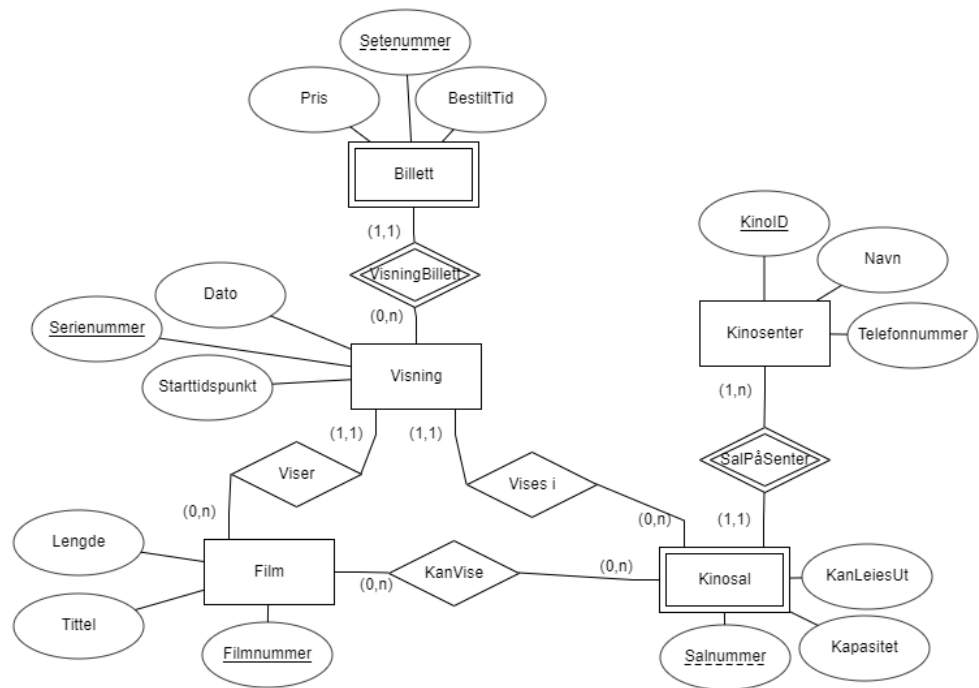
b) Hvis kardinaliteten endres til (0,1), vil den ikke kunne være en svak klasse siden da kan en kinosal eksistere uten at den er koblet til et kinosenter. Kardinaliteten kan heller ikke være (1,n) ettersom en kinosal ikke kan eksistere i flere kinosenter. En unik kinosal kan kun eksistere i et kinosenter så derfor må kardinaliteten strengt være (1,1).

c) Antagelser: Det er gitt gjennom relasjonen at en film kan vises i en sal, så salen har f.eks 3D-funksjonaliteten som trengs dersom den kan vise filmen.

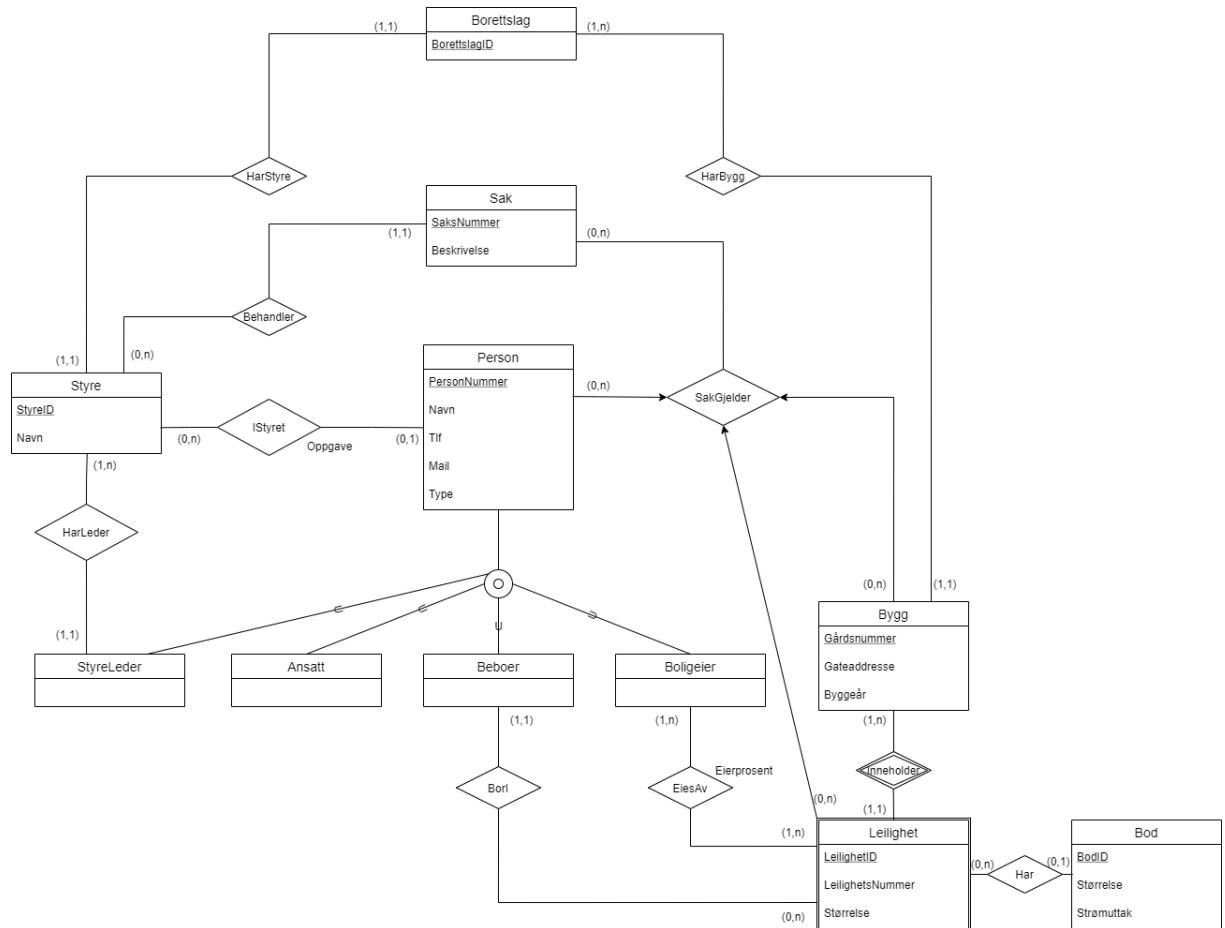


d)

e) Antagelser: En visning viser kun en Film, en visning har kun en kinosal, alle sete-nummerene i en sal er unike



4 Fra miniverden til ER-modell



Antagelser

- Antar at en beboer kun kan bo i en leilighet derav (1,1)
- Antar at en boligeier må eie en leilighet for å være boligeier derav (1,n)
- Antar at det kan eksistere boder som ikke tilhører en leilighet derav (0,1)
- Antar at en bod kun kan "eies" av en leilighet og ikke deles derav (0,n)

Angående SakGjelder relasjonen ønsker vi å vise at sak kan være knyttet til en vilkårlig kombinasjon av entitetene, vi ønsker i utgangspunktet ikke å knytte for eksempel kun Person og Leilighet via SakGjelder relasjonen, men vi er litt usikker på hvordan vi skal uttrykke dette.