TDT4145 - Øving 1

Sander Lindberg

1 Oppgave 1: Databasesystemer

1.1 Oppgave a

> *"Forklar kort og overordnet hva en database (DB) og et databasehåndteringssystem (DBMS) er."*

En database er en strukturert samling med data, som er lagret i en pc/server. Dataen er en samling av relaterte dataer. F.eks kan du lagre informasjon om en person, som navn, nummer, fødselsdato, barn, foreldre, kjæledyr osv...

Databasehåndteringssystem eller DBMS er et program en bruker for å håndtere databaser. For eksempel **DB2**, **Oracle**, **MySQL** osv...

1.2 Oppgave b

> *"En database og et databasehåndteringssystem kan man tenke seg til sammen utgjør et databasesystem. Det er mange fordeler ved å bruke et databasesystem i motsetning til tradisjonelle filsystemer der hver brukergruppe selv tar hånd om egne filer for å behandle lagring av data. Forklar nærmere hva som menes med følgende tre egenskaper og hvilke fordeler det medfører:"*

1. **Program-data uavhengighet:** Sørger for at dataen fortsatt har integritet og tilgjenglighet, selvom databasen blir endret, enten fysisk eller logisk.

2. **Flerbrukersstøtte:** Går ut på at databasen støtter at flere kan aksessere og endre dataen samtidig. For eksemel om to stykker skal oppdatere en saldo samtidig, da vil kun det siste innskuddet til saldoen bli registrert om databasen ikke har flerbrukerstøtte, mens begge vil registreres dersom den gjør.

3. **Selvbeskrivende:** Hvis databasen inneholder en beskrivelse av sin egen struktur, sies den å være *selvbeskrivende*.

2 Oppgave 2: ER-modellen

2.1 Oppgave a

Forklar:

1. **Forskjellen på en entitet og en entitetsklasse:** En **entitet** er et objekt av en **entitetsklasse**. Entitetsklassen er en "oppskrift" på hvordan en entitet skal se ut. For eksempel kan en entitetsklasse være (igjen) Person, som har attributtene "ID", "Navn", "Fødselsdato" osv...Mens en spesiell person (entitet) kan ha fylt inn disse attributtene med data; "ID: 1", "Navn: Sander Lindberg", "Fødselsdato: 27.06.98".

2. **Forskjellen på en relasjon og en relasjonsklasse:** En **relasjon** er en sammenheng mellom to eller flere entiteter, f.eks mellom en Person og en Hund. **Relasjonsklassen** her kan være "Eier". Kan også ha rekursive relasjoner, som er relasjoner mellom like entiteter, f.eks Person og Person. En person kan være sjef for en annen person, og denne personen kan være arbeidstaker av sjefen. Relasjonsklasser kan også ha attributter. F.eks en relasjonsklasse "BittAv" (som sier om en person er bitt av en hund eller ikke) kan ha attributten "Antall", som sier hvor mange ganger.

3. **Hvorfor alle entiteter må ha et eller flere nøkkelattributt:** For å kunne identifisere de. Nøkkelattributtene er unike for hver entitet, f.eks en ID eller et fødselsnummer for en person.

2.2 Oppgave b

"Betrakt ER-modellen under og avgjør om følgende utsagn kan stemme. For hvert utsagn skal du svare "true", "false" eller "maybe" og gi en kort begrunnelse for svaret ditt. Fyll svarene dine inn i tabellen

1. *Taco er en entitetsklasse og hver enkelt instans av en taco identifiseres ved hjelp av nøkkelattributtet TacoID.*

2. *En taco kan inneholde et vilkårlig antall kjøttdeiger, oster, grønnsaker og sauser.*

3. *En ordre gjort av en kunde trenger ikke inneholde noen taco.*
4. *En ordre gjort av en kunde kan inneholde en million tacoer.*
5. *Så fort en ordre er opprettet kan den hentes i en vilkårlig butikk.*
6. *En kunde kan eksistere i systemet uten noensinne å ha bestilt noen tacoer.*
7. *Hver eneste grønnsak-entitet har en vekt.*
8. *En ansatt kan jobbe i ulike butikker med ulike stillingstitler*
9. *Alle ordre delegeres typisk til denne samme personen som er ansatt.*
10. *En kunde må være registrert med et navn.*

”

(Videre er tallet hvilken påstand som besvares)

1. **True**. Taco er en firkant, som er notasjonen for entitetsklasse,i tillegg vil alle instsanser av taco ha samme egenskaper. Oppfyller også kravet om en unik identifikator; TacoID. TacoID er understreket, som sier at det er en nøkkel.
2. **True**. Alle relasjonene er (0, n), som sier at Taco **kan** ha vilkårlig mange kjøttdeiger, sauser, oster og grønnsaker, men *trenger* ikke ha noen.
3. **False**. Relasjonen bestilling mellom ordre og taco har en restriksjon (1, n), som sier at den **må** ha en taco, og kan ha vilkårlig mange.
4. **True**. Svart på i forrige punkt.
5. **False**. Det er en (1, 1) restriksjon mellom ordre og butikk, som sier at en ordre **må** ha en og bare en butikk.
6. **True**. Restriksjonen (0, n) mellom kunde og en ordre sier at en kunde **trenger** ikke ha en ordre for å være kunde.
7. **Maybe**. En grønnsak har en kilopris, som impliserer at den har en vekt, men det står ingenting om hva vekten er. Vekten kan ligge i beskrivelse.
8. **True**. Restriksjonen (1, n) sier at en ansatt **må** ha en jobb (relasjonsklassen mellom ansatt og butikk, som har attributten stillingstittel), og at han kan ha uendelig mange

jobber. Han vil dog ha en annen AnsattID hos de forskjellige butikkene, da denne er unik (strek under).

9. **Maybe**. Altså, denne ansatte kan ha utrolig uflaks og bli delegert alle ordre, men restriksjonen sier at den kan gå til uendelig mange, så alle ansatte kan få den. En ansatt har dog en ”Tilgjengelighet”-attributt, som jeg antar sier om han allerede er delegert en ordre eller ikke, så det kan jo hende det er noe greier der som passer på at en ansatt ikke blir delegert alle ordre.

10. **Maybe**. Dunno om ”Navn”-attributtet er satt til NotNull eller ikke. Om den er satt til det, må en kunde ha navn.

3 Oppagave 3: Svake klasser, forekomstdiagram og nye krav

3.1 Oppgave a

> *”Over ser du et eksempel på en svak klasse. Når kan det være hensiktsmessig med svake klasser? Forklar hva i modellen over som er identifiserende entitetsklasse, identifiserende relasjonsklasse og delvis nøkkel.”*

Det kan være hensiktsmessig med svake klasser dersom en entitetsklasse mangler en ”naturlig” nøkkel. Kinosenter er identifiserende entitetsklasse. Denne er ”eieren” av kinosalen. Den identifiserende relasjonsklassen er ”SalPåSenter” og den delvise nøkkelen er ”Salnummer” (striplet underline)

3.2 Oppgave b

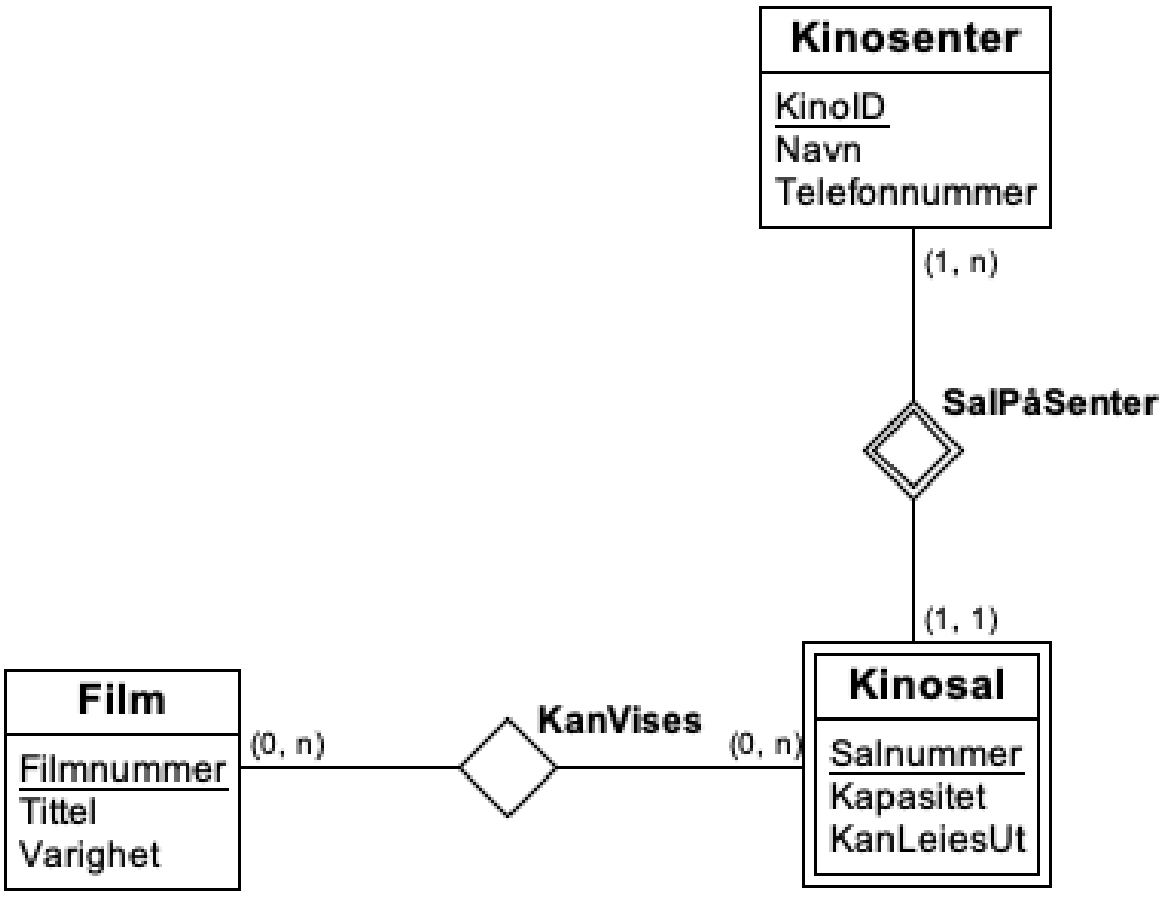
> *”Hva skjer hvis kardinaliteten fra Kinosal til Kinosenter er (0,1) eller (1,n)? Kan vi fortsatt modelle Kinosal som svak? Hvorfor eller hvorfor ikke?”*

Hvis den er (0, 1), sier det at Kinosal ikke trenger noe Kinosenter, (1, n) sier at den kan ha uendelig mange, men må ha en. I begge tilfellene må vi opprette en ”vanlig” (ikke delvis) nøkkel i Kinosal og da er det ikke en svak entitetsklasse lenger. Grunnen til at vi må opprette nøkkel i det føreste tilfelle er fordi en svak Kinosal ikke kan eksistere uten et Kinosenter og

det andre tilfelle må vi vite hvilken kinosal det er snakk om, så vi ikke ser på en kinosal fra et annet kinosenter (da den kan ha flere, (1, n)).

3.3 Oppgave c

"Utvid ER-modellen til å ta høyde for at en sal kan godkjennes for et vilkårlig antall filmer. Det er nemlig ikke alle saler som kan vise alle filmer. For eksempel krever 3D-filmer spesielle skjermer som kun finnes i noen saler. En film har et unikt filmnummer, en tittel og en varighet. En film kan ikke ha noen begrensning i antall saler den kan godkjennes for. Skriv ned eventuelle antagelser du finner det nødvendig å gjøre."



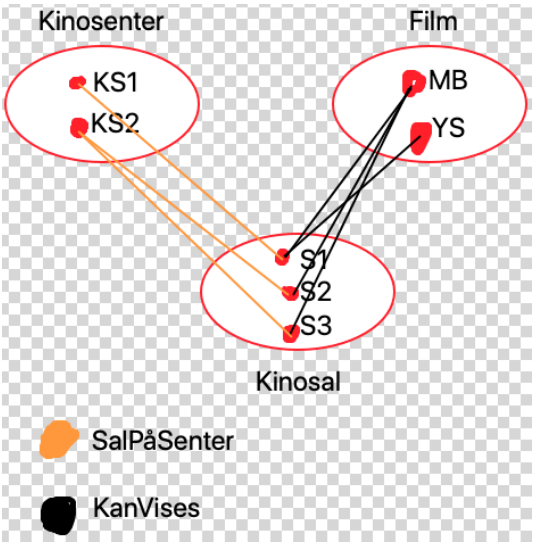
Figur 1: Utvided ER-modell

3.4 Oppgave d

Anta at vi nå har en database realisert med ER-diagrammet ditt og med følgende data:

- *To kinosentere (KS1 og KS2).*
- *KS1 har kun en sal (S1), KS2 har to saler (S2 og S3).*
- *Vi har to filmer i databasen: Matban Begins (MB) og Yes Summer (YS). MB er godkjent for alle de tre salene, YS er kun godkjent for S1.*

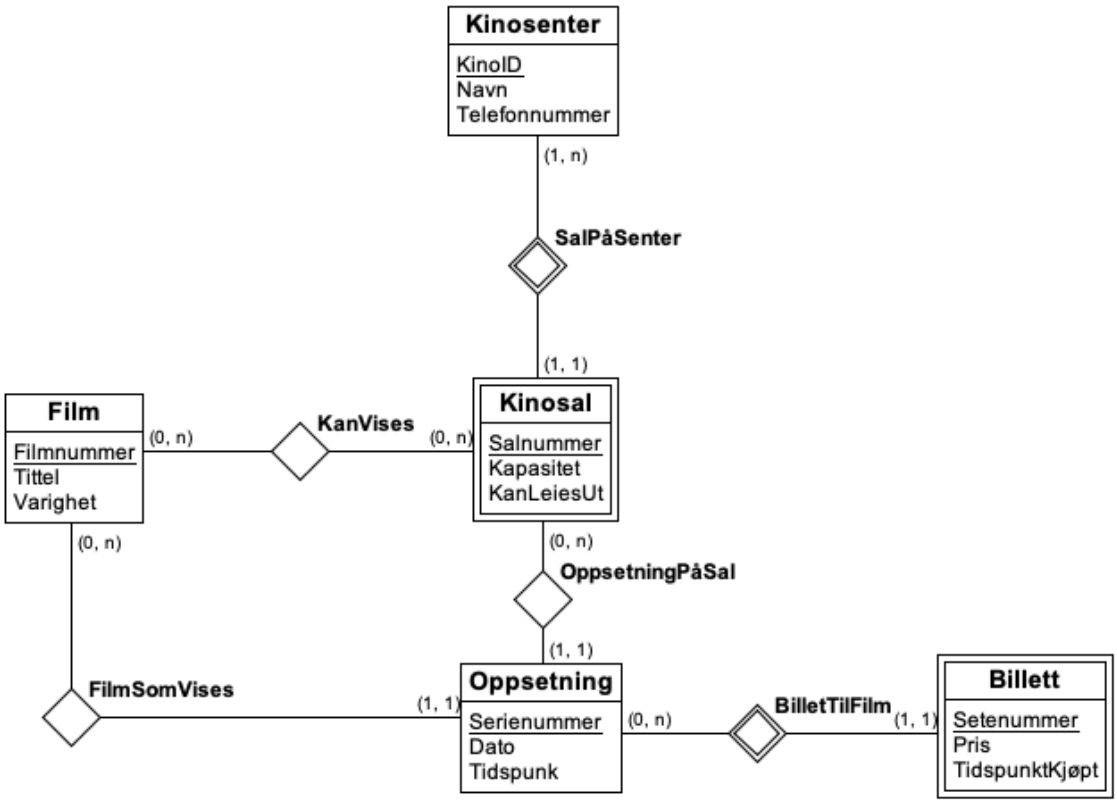
Tegn et forekomstdiagram for denne databasetilstanden.



Figur 2: Forekomstdiagram oppgave 3d

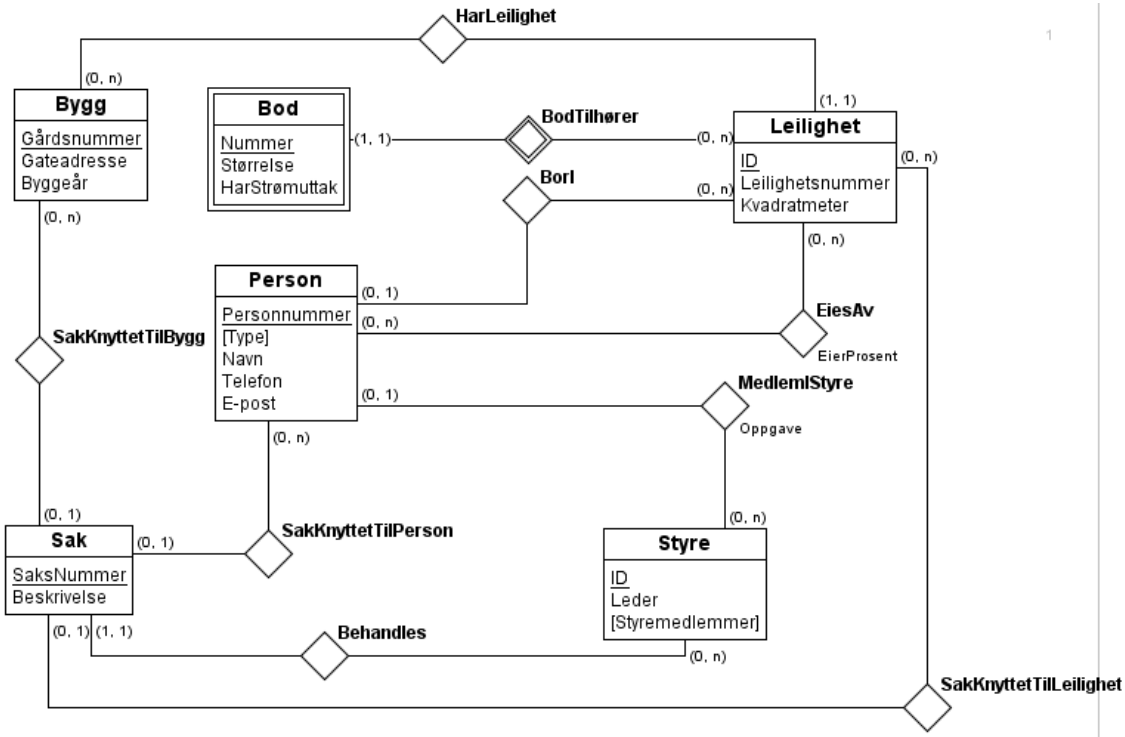
3.5 Oppgave e

Utvid modellen til å ta høyde for oppsett av filmfremvisninger og tilhørende billetter: En oppsetning av en film er knyttet til en bestemt kinosal og må ha en dato og et tidspunkt for når filmen starter. Hver oppsatt film skal få et eget serienummer som er unikt blant alle kinosentere i systemet. Kinosenteret ønsker også å vite hvor mange billetter som selges og hvor mye som tjenes til en bestemt fremvisning. En billett til en oppsatt film trenger et setenummer og har en pris. Det skal også registreres hvilket tidspunkt billetten ble kjøpt. Skriv ned eventuelle antagelser du finner det nødvendig å gjøre.



Figur 3: Utvided ER-modell oppgave 3d

4 Oppave 4



Figur 4: Oppgave 4 - Fra miniverden til ER-modell

Jeg har laget en ER-modell av miniverdenen. Jeg har satt ”Bod” som en svak klasse, da jeg tenker den ikke kan eksistere uten en leilighet og id’en skal være unik innenfor hver leilighet (selvom bod med ID 1 kan finnes i både leilighet 1, 2, 3 osv...). Når det kommer til strømuttak i boden, satt jeg en ”boolean” attributt, da denne kan være true eller false.

For eierskap av leilighet satt jeg en attributt i relasjonen ”EiesAv” for antall eierprosent. Leiligheten kan eies av 0 eller mange *personer*. Ser at denne kan være litt far fetched, men leiligheten *kan* være eid av kommunen/staten (hva vet jeg) og det er ingen person.

En person trenger heller ikke eie en leilighet, så det blir en (0, n) restriksjon her, da en person også kan eie så mange leilgheter han vil. Jeg tenkte også at en person ikke kan bo i mer enn en leilighet samtidig, derfor satt jeg en restriksjon på (0, 1) i ”BorI” relasjonen, da en person ikke *trenger* å bo i en leilighet, men kan bo i max 1. En leilighet kan ha (i følge oppgaveteksten, veldig urealistisk) uendelig mange beboere, men trenger ikke ha noen, derfor en (0, n) der.

En person skulle også kunne ha en ”type”, jeg satt denne som en fler-verdi attributt, da det kan være mange forskjellige typer. Samme gjelder ”Styremedlemmer” for styret.

Saker skulle kunne være knyttet til bygg, person og/eller leilighet og skulle behandles av styret. Jeg tenkte styret kunne behandle flere saker samtidig, så satt en (0, n) restriksjon der. En sak *må* bli behandlet av styret og det er kun et styre, så (1, 1). Ellers satt jeg relasjoner mellom sak og bygg, sak og person og sak og leilighet, for å gjøre det mulig for en sak å være tilknyttet de forskjellige.