

Oppgave 1: Databasesystemer

- a) En **database** er en organisert samling av relaterte data, som representerer fakta med implisitt mening. Disse dataene reflekterer en del av den virkelige verden, ofte referert til som en "miniverden", og endringer i denne miniverdenen speiles i databasen. Hensikten med en database er å lagre informasjon på en slik måte at den kan hentes og oppdateres effektivt. Størrelsen på en database varierer, og den kan omfatte alt fra en enkel telefonbok til komplekse strukturer som finnes i store organisasjoner eller sosiale nettverksplattformer.

Et **databasehåndteringssystem** er et programvaresystem som tillater brukere å definere, opprette, manipulere og dele databaser. Definering av en database innebærer å spesifisere datatyper, strukturer og begrensninger. Å opprette en database innebærer å lagre data på en lagringsplass kontrollert av DBMS. Manipulasjon av databasen omfatter funksjoner som å utføre spørringer for å hente spesifikke data, oppdatere databasen for å reflektere endringer, og generere rapporter. Deling av databasen gjør det mulig for flere brukere og applikasjoner å få tilgang til og samhandle med databasen samtidig.

b) Databasesystem

- 1) **Program-data uavhengighet:** Her mener vi at strukturen til dataene lagres uavhengig av programmene som benytter det. I et databasesystem er dataene sentralisert og ikke innbakt i programmet. Dette betyr at man kan endre databasens struktur uten å måtte gjøre endringer i programmene som bruker disse dataene.
- 2) **Flerbrukerstøtte:** Et databasesystem er designet for å støtte at flere brukere arbeider med data samtidig. Må legge til en mekanisme for å håndtere evt. tilfelle der to brukere vil endre samme data samtidig.
- 3) **Selvbeskrivende:** I et databasesystem lagres ikke bare selve dataene, men også detaljert informasjon om dataene (metadata). Dette innebærer at systemet inneholder en beskrivelse av hva slags data det inneholder, hvordan dataene er strukturert, og hvilke typer data som finnes. Dette gjør at systemet ikke trenger ekstra informasjon for å forstå innholdet og strukturen til dataene det håndterer. Fordeler med dette er brukervennlighet og integrering med nye applikasjoner

Oppgave 2: ER-modellen

a) Forklar

- 1) En entitetsklasse er en «mal» og inneholder attributter, og entitet er et konkret eksempel på denne malen.
- 2) En relasjonsklasse sier hva relasjonen kobler, og relasjon er et konkret eksempel.
- 3) Alle entiteter må ha et eller flere nøkkelattributt for å kunne identifisere og skille mellom entitetene i samme klasse.

b)

Påstand nr.	Svar	Begrunnelse
1	True	Definisjonen
2	True	Øvre begrensing er n (0, n)
3	False	En ordre må inneholde minst en taco (1, n)
4	True	Øvre begrensing er n og n er ukjent
5	False	Kan bare hentes i en bestemt butikk (1, 1)
6	True	Trenger ikke å ha lagt inn noen ordre (0, n)
7	Maybe	Hver grønnsak som skal i TACOEN har en vekt
8	True	Øvre begrensing er n (0, n)
9	Maybe	Hver ordre blir delegert til en ansatt, men hvilken ansatt kommer an på butikk og hvor mange som jobber der
10	Maybe	Kundeklassen har en navn-attributt, men står ingenting om at den MÅ fylles

Oppgave 3: Svake klasser, forekomstdiagram og nye krav

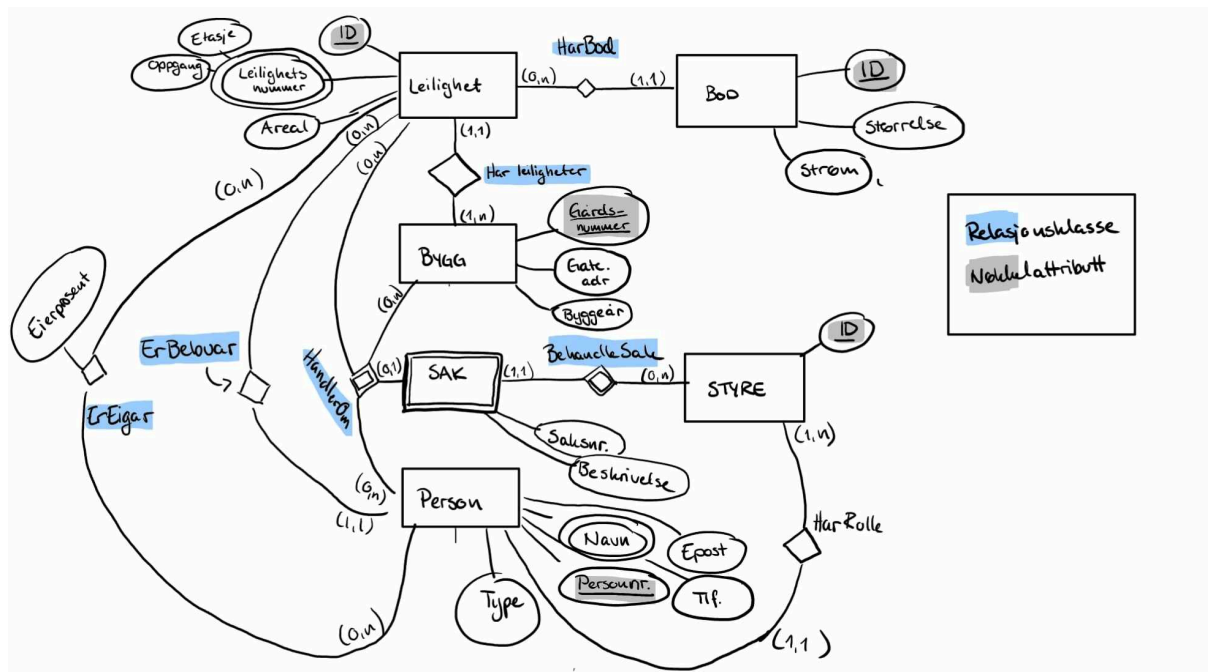
- a) Det kan være hensiktsmessig med svake klasser når entitetsklassen ikke har en naturlig identifikator. Dette gjør at vi ikke trenger et ekstra nøkkelattributt, og sparer oss dermed for ekstra arbeid.

Identifiserende entitetsklasse: Kinosenter

Identifiserende relasjonsklasse: SalPåSenter

Delvis nøkkel: Salnummer

Oppgave 4: Fra miniverden til ER-modell



Antakelser:

Siden alle har ulike personnummer vil denne være unik og kunne settes som nøkkelattributt.

Setter gateadresse som enkel attributt og ikke flerverdi da det ofte kun er en verdi som skiller dem(A-Å)

Et bygg må ha et minimum antall leiligheter for å være et bygg. Altså vil vår modell ikke godta et bygg uten leiligheter.

Det er antatt at en sak kun vil ta for seg en leilighet eller en person, ikke to personer.