

## Fag TDT4145 Datamodellering og databasesystemer

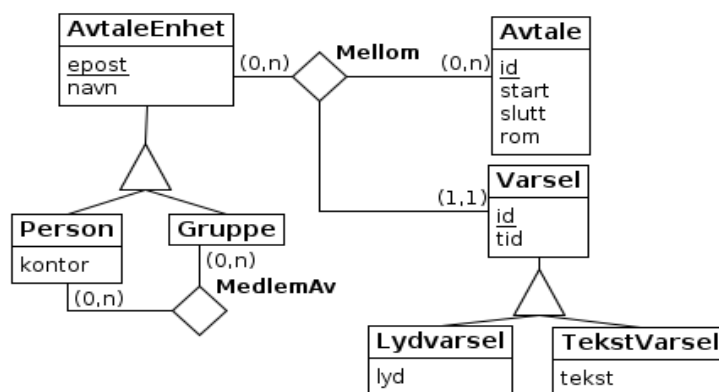
### Løsning til øving 1: Introduksjon til databaser og ER-modellering

#### Oppgave 1

- α) Med data menes kjente resultater eller fakta som har implisitt mening og kan lagres. En database er definert som en samling av relaterte data. Selve databasen og all programvare for å manipulere databasen kalles til sammen et databasesystem.
- β) Ved bruk av tradisjonelle filsystemer, implementerer hver bruker de funksjoner og filer han trenger for hver applikasjon. Ved bruk av databaser defineres databasen en enkelt gang, og man holder en enkelt samling av alle data som deretter kan brukes av flere brukere og applikasjoner. Noen viktige hovedpunkter er ellers:
- Datakatalogen.
  - Skille program og data.
  - Dataabstraksjoner.
  - Flere synspunkt (view) av samme database.
- c) Fordeler ved å bruke databasesystemer framfor vanlige filsystem er flere:
- Kontrollert redundans.
  - Deling av data.
  - Adgangskontroll.
  - Kontroll av integritet.
  - Samtidighetskontroll.
  - Sikkerhetskopiering , arkivering og reetablering.
  - Mulighet for standardisering.
  - Fleksibilitet.
  - Redusert utviklingstid.
  - Tilgjengelige og oppdaterte("up to date") data.
  - Storskala drift.

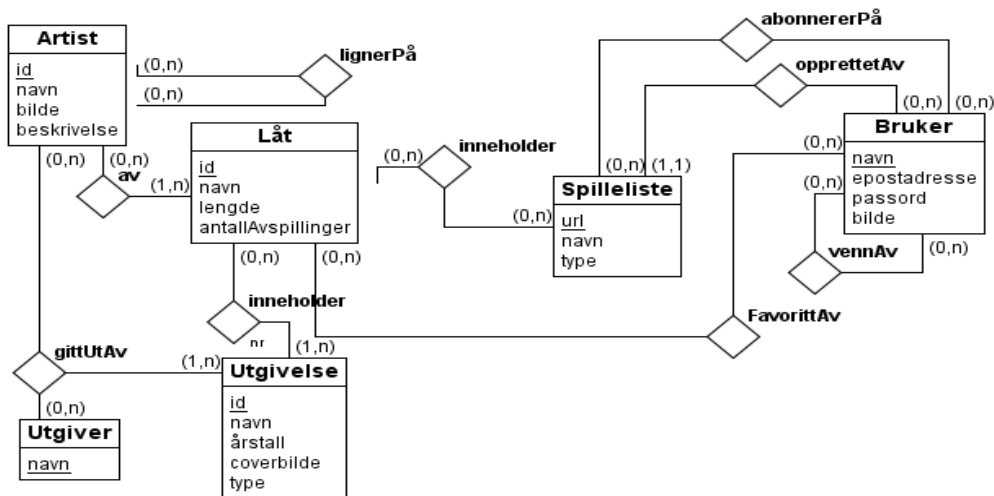
Det er verdt å merke seg at det ofte medfører store kostnader å anskaffe og ta i bruk et stort databasehåndteringssystem, slik at det ikke alltid vil være den beste løsningen. For små systemer og for enbrucker systemer, kan et filsystem-basert konsept være mer hensiktsmessig. Et databasehåndteringssystem (DBMS) medfører også en relativt stor "overhead" som kan være et problem i tidskritiske applikasjoner.

## Oppgave 2



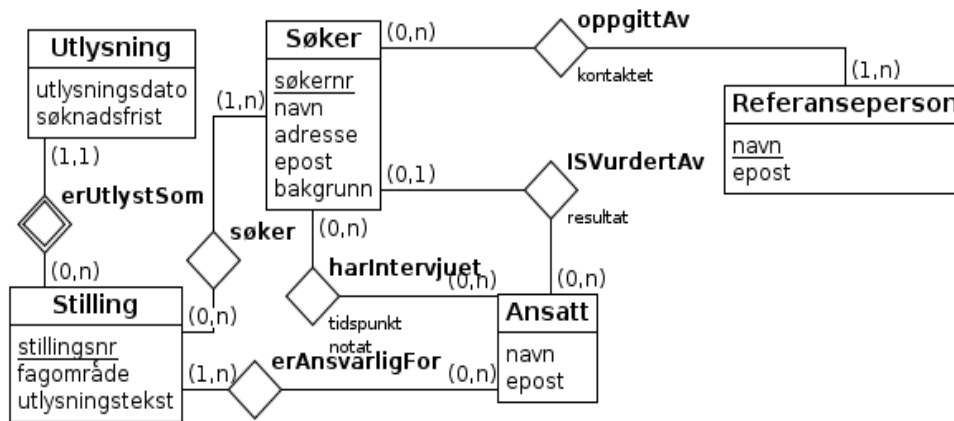
Her har vi valgt å modellere *AvtaleEnhet* som et Isa-hierarki, slik at *Mellom* kan defineres for begge typene avtaleenheter. *Avtale* er en entitet, og ikke en relasjon, slik at vi fritt kan ha avtaler med kun en deltager (f.eks., personen er utilgjengelig). *Varsel* er også et hierarki, men det er ikke så viktig. Vi har systemgenererte nøkler for *Avtale* og *Varsel*, og epost er en nøkkel for alle avtaleenhetsklasser. Ukentlige avtaler modelleres ikke spesielt, vi antar det er måten de settes inn på (og evt. fjernes) som er spesiell.

## Oppgave 3



Her er det mange designmuligheter. Vi har valgt å la representere ulike varianter *Utgivelser*, med ett *type*-attributt. I så fall har vi ingen eller flere *Artist* og eksakt en *Utgiver* stående bak hver utgivelse. Siden navn til verken *Låt*, *Artist*, *Utgivelse* eller *Spilleliste* skal være unike, bruker vi en ekstra identifikator for de tre første og en URL den siste. Ulike typer av spillelister håndterer vi også med et *type*-attributt. Alternativt kunne vi bruke ISA for både *Utgivelse* og *Spilleliste*, men i vårt tilfelle ville det bare øke kompleksiteten til modellen uten å gi noen spesielle fordeler. Endelig representerer vi kobling mellom *Spilleliste* og *Bruker* som to forskjellige relasjoner, *abonnertPå* og *opprettetAv*. Hver bruker kan opprette og abonnere på ingen eller mange spillelister. Hver liste, derimot, kan abonneres av ingen eller mange brukere, mens den er opprettet av en og bare en bruker.

## Oppgave 4



Det er igjen mange ulike måter å lage ER-diagram på. I vårt eksempel velger vi Utlysning, Stilling, Søker, Ansatt og Referanseperson være framstilt som egne entiteter. Relasjonen mellom Søker og Referanseperson har et eget attributt, kontaktet, som tenkes å si hvorvidt en oppgitt referanseperson har vært kontaktet eller ikke. IS-vurdering og intervju representerer vi som relasjoner mellom Ansatt og Søker, hvor den første har et resultat som attributt, mens den andre har et tidspunkt og en notat. Endelig velger vi å ha en relasjon mellom Stilling og Ansatt for å beskrive hvem er den ansvarlige for den aktuelle stillingen, her velger vi også å ha 1 til mange ansvarlige for enhver utlyst stilling.