IN2090 – Databaser og datamodellering

04 – Eksempeloppaver: Modellering og realisering

Leif Harald Karlsen
leifhka@ifi.uio.no



Universitetet i Oslo

Oppgave 1 – Ternære relasjoner

Vi vil lage et ER-diagram som modellerer tillatelser. Lag ER-modellen som fanger følgende informasjon:

1. En godkjenning er noe som gis til en person av en autoritet for en gitt tillatelse. Vi har altså entitetene *person*, *autorietet* og *tillatelse*, og en relasjon *godkjenning* mellom disse.
2. En person identifiseres med et personnummer, en autoritet identifiseres med et navn og en tillatelse identifiseres med et navn.
3. En person kan få mange tillatelser fra en autoritet;
4. en autoritet kan gi mange personer samme tilatelse;
5. men kun én autoritet kan gi en bestemt tillatelse til en bestemt person.

Oppgave 2 – Ternære relasjoner

Vi vil lage et ER-diagram som modellerer salg av varer fra butikker til kunder. Lag ER-modellen som fanger følgende informasjon:

1. Varer selges av en butikk til en person. Vi har altså entitetene *butikk*, *kunde* og *vare*, og en relasjon *selger* mellom disse.
2. En kunde identifiseres med et kundenummer, en vare identifiseres med en strekkode og en butikk identifiseres med en adresse.
3. En kunde kan kjøpe samme vare fra mange butikker;
4. en butikk kan selge mange varer til samme kunde;
5. en butikk kan selge samme vare til mange kunder.
6. En kunde må ha kjøpt minst én vare fra en butikk.

Oppgave 3 – En mer realistisk oppgave

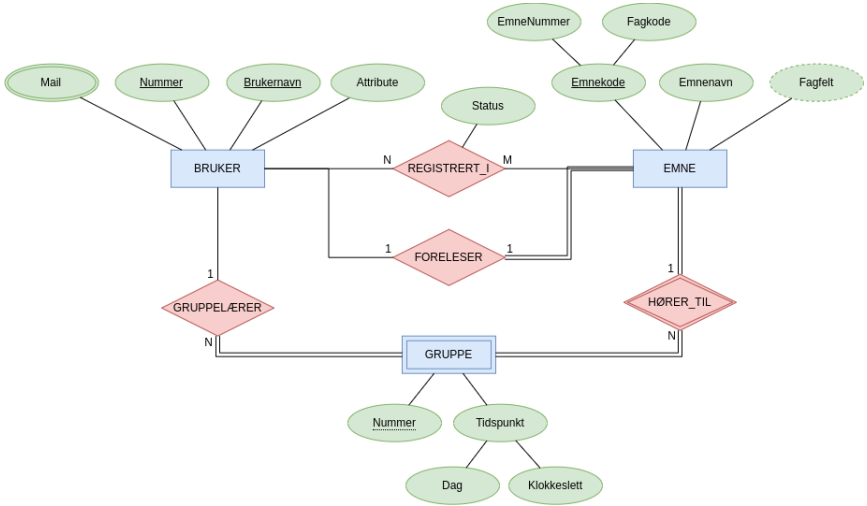
Lag en modell for et biblioteksystem, hvor du får oppgitt følgende (tvetydige) informasjon fra ulike ansatte:

1. Ansatt 1 sier: ”Biblioteket har bøker som skal lagres i systemet. Hver bok har en tittel, en forfatter-streng (f.eks. ’Elamsri & Navathe’) og et unikt ISBN-nummer.”
2. Ansatt 2 sier: ”For en bok er kombinasjonen av tittel og forfatter unikt.”
3. Ansatt 3 sier: ”Biblioteket kan ha flere av samme bok. Hver av disse kan være på utlån i en periode og har et unikt serienummer.”
4. Ansatt 4 sier: ”En bok kan komme i mange utgaver, slik som 1. utgave, 2. utgave, osv., men også feks. internasjonal utgave. ISBN-nummeret er da forskjellig for hver utgave, mens tittel og forfatter-streng er lik. Utgaver har også en dato de er utgitt. Den første utgaven av hver bok kaller vi bare ’original-utgaven’.”

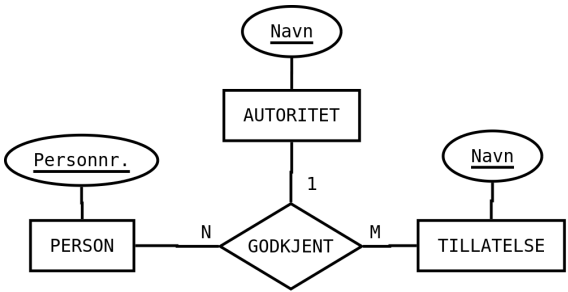
Hint: Kan ordet ”bok” knyttes til den samme entitets-typen i alle setningene over?

Oppgave 4 – Realisering

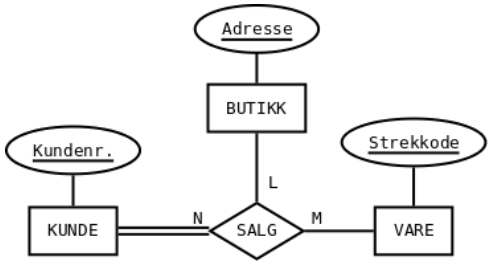
Realiser følgende ER-diagram til et databaseskjema.



Oppgave 1 – Løsning

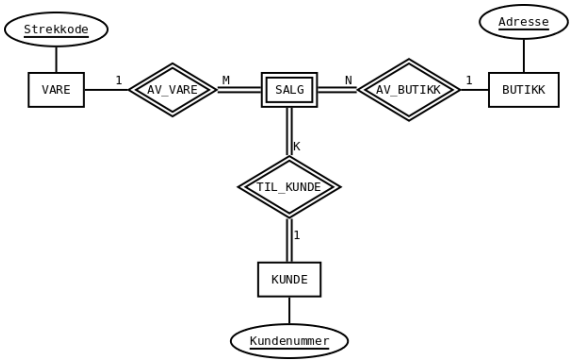


Oppgave 2 – Løsning

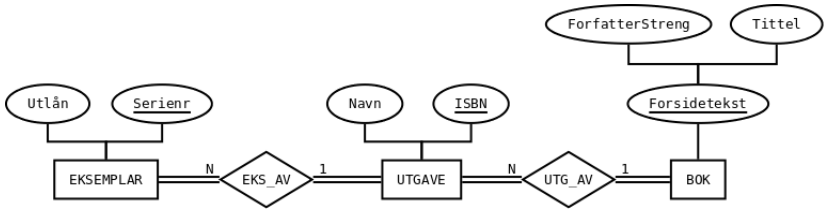


Oppgave 2 – Alternativ øsning

Alternativ løsning med svak entitet og binære relasjoner i stedet for ternær relasjon.



Oppgave 3 – Løsning



Oppgave 4 – Løsning

Bruker her enkel understrek for å markere kandidatnøkler og dobbel understrek for å markere primærnøkkel.

- Starter med å lage en relasjon per (ikke-svake) entitet (hvor vi ser bort ifra multi-verdi attributtet `Mail` og det utledbare attributtet `Fagfelt`) og må velge en av nøklene som prmiærnøkkel (dobbel-understreket dersom det er fler kandidatnøkler):

  `Bruker(nummer, brukernavn, navn)`

  `Emne(emnenummer, fagkode, emnenavn)`
- Så vil vi realisere svake eniteter. Her har vi kun `GRUPPE`. For denne må vi i tillegg til attributtet `Nummer` ha nøkkelattributtene til dens eiende entiteter, som her kun er `EMNE`. Nøkkelen blir så kombinasjonen av dens svake nøkkel og nøklene til dens eiende entiteter. I tillegg har vi de andre attributtene, altså får vi:

  `Gruppe(nummer, emnenummer, fagkode, dag, klokkeslett)`

  hvor `nummer` er den svake nøkkelen og `(emnenummer, fagkode)` refererer til `Emne(emnenummer, fagkode)`.

Oppgave 4 – Løsning (fort.)

- Deretter må vi realisere de vanlige relasjonene. Vi starter med `FORELESER_I`-relasjonen. Denne er 1-1. Siden hvert emne må ha en foreleser, gir det mest mening å legge denne relsjonen inn som en attributt i `Emne`-relasjonen. Vi oppdaterer derfor denne tabellen slik:

  `Emne(emnenummer, fagkode, emnenavn, foreleser)`

  hvor `foreleser` refererer til `Bruker(nummer)`. Merk at `foreleser` nå blir en kandidatnøkkel, siden en bruker bare kan forelese ett emne (og alle emner har en foreleser).

- Så har vi `GRUPPELÆRER_I` som er en 1-N-relasjon. Her kan vi velge om vi vil ha den i en egen relasjon, eller om vi vil putte relasjonen inn i `Gruppe`-relasjonen. Siden hver gruppe må ha en gruppelærer gir det mening å putte denne inn i `Gruppe`-relasjonen, og den vil da se slik ut:

  `Gruppe(nummer, emnenummer, fagkode, dag, klokkeslett, gruppelærer)`

  hvor `Gruppe(gruppelærer)` refererer til `Bruker(nummer)`.

Oppgave 4 – Løsning (fort.)

- Videre har vi `REGISTRERT_I`, som er en M-N-relasjon, så må derfor få sin egen relasjon:

  `Registrert_i(nummer, emnenummer, fagkode, status)`

  hvor `Registrert_i(nummer)` refererer til `Bruker(nummer)` og `Registrert_i(emnenummer, fagkode)` refererer til `Emne(emnenummer, fagkode)`.

Oppgave 4 – Løsning (fort.)

- Til slutt må vi realisere multi-verdi attributter. Den eneste slike er `BRUKERs Mail`. Vi lager da en egen relasjon som inneholder en attributt `nummer` som peker på `Bruker(nummer)` samt en attributt `mail` som inneholder mailadressen:

  `Mail(nummer, mail)`

Oppgave 4 – Løsning (fort.)

Hele skjemaet blir da:

Bruker(nummer, brukernavn, navn)
Emne(emnenummer, fagkode, emnenavn, foreleser)
Gruppe(nummer, emnenummer, fagkode, dag, klokkeslett, gruppelærer)
Registrert_i(nummer, emnenummer, fagkode, status)
Mail(nummer, mail)

hvor vi har fremmednøklene:

Gruppe(gruppelærer) → Bruker(nummer)
Emne(foreleser) → Bruker(nummer)
Registrert_i(emnenummer, fagkode) → Emne(emnenummer, fagkode)
Registrert_i(nummer) → Bruker(nummer)
Mail(nummer) → Bruker(nummer)

Oppgave 4 – Løsning (alternativ syntaks)

Her er samme databaseskjema skrevet med en alternativ syntaks. Her er ‘KN‘ kort for kandidatnøkler og ‘PN‘ kort for primærnøkler.

```
# Relasjoner

Bruker(nummer, brukernavn, navn)
- KN: {nummer}, {brukernavn}
- PN: {nummer}
Emne(emnenummer, fagkode, emnenavn, foreleser)
- KN: {emnenummer, fagkode}, {Foreleser}
- PN: {emnenummer, fagkode}
Gruppe(nummer, emnenummer, fagkode, dag, klokkeslett, gruppelærer)
- KN: {nummer, emnenummer, fagkode}
- PN: {nummer, emnenummer, fagkode}
RegistrertI(nummer, emnenummer, fagkode, status)
- KN: {nummer, emnenummer, fagkode}
- PN: {nummer, emnenummer, fagkode}
Mail(nummer, mailadresse)
- KN: {nummer, mailadresse}
- PN: {nummer, mailadresse}

# Fremmednøkler:

Gruppe(emnenummer, fagkode) -> Emne(emnenummer, fagkode)
Emne(foreleser) -> Bruker(nummer)
Gruppe(gruppelærer) -> Bruker(nummer)
RegistrertI(nummer) -> Bruker(nummer)
RegistrertI(emnenummer, fagkode) -> Emne(emnenummer, fagkode)
Mail(nummer) -> Bruker(nummer)
```