

# IN2090 – Repetisjonsoppgaver

## Modellering

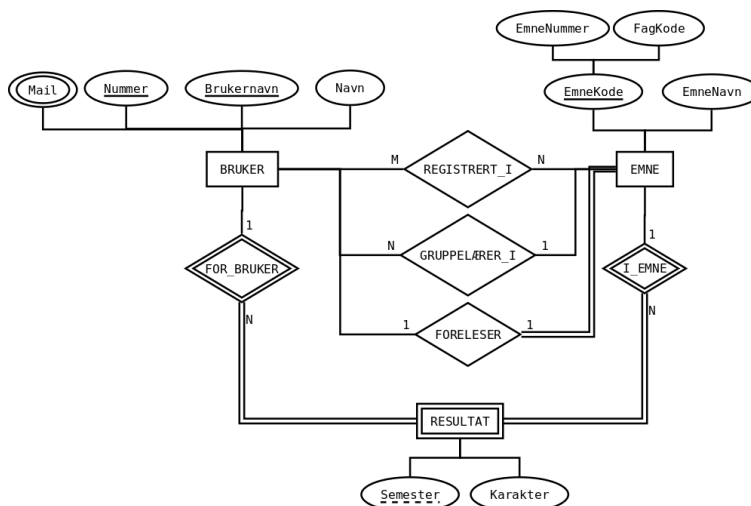
### Oppgave 1 – ER-modellering

a)

(Inspirert av Oppg 2.1 fra Eksamen 2018) Lag en ER-modell som inneholder denne informasjonen om brukere, emner og resultater:

1. Hver bruker er identifisert av et unikt nummer (f.eks. 483226). I tillegg skal alle brukere ha et brukernavn (f.eks. har bruker 483226 brukernavnet "olanor"). Ingen brukere kan ha samme brukernavn. Vi vil også registrere fullt navn for brukerne, men dette vil kanskje mangle for noen brukere. Flere brukere skal kunne ha samme navn. En bruker kan også ha mange emailadresser.
2. Brukere kan være registrert i flere emner. Emner er representert av en emnekode som består av to deler, et emnenummer (f.eks. "1010") og en fagkode (f.eks. "IN"). I tillegg skal alle emnene ha et emnenavn, og flere emner kan ha samme navn. Det er ingen øvre grense på hvor mange brukere som kan være registrert i et emne.
3. Noen brukere er gruppelærer for emner. En bruker kan kun være gruppelærer for ett emne, men hvert emne kan ha mange gruppelærere.
4. En bruker kan være foreleser i et emne (men ikke flere). Et emne må ha nøyaktig én foreleser.
5. Vi vil også registrere resultater fra tidligere semestre, slik at hver bruker kan ha ett resultat per emne for et gitt semester. For eksempel kan bruker 483226 ha fått karakteren "C" i emnet "(IN,1010)" i semesteret "vår 2017". Det skal være mulig å registrere resultat for samme emne i flere semestre, f.eks. kan også bruker 483226 få registrert "B" i emnet "(IN,1010)" vår 2018. Det er mulig for en bruker å ha samme karakter i flere semestre, men en bruker kan ikke ha flere resultater i et emne i samme semester. Semestre representeres med en semesterkode (f.eks. "vår 2018"). Resultat kan være "bestått" eller "ikke bestått", eller en bokstav mellom "A" og "F".

## Løsningsforslag

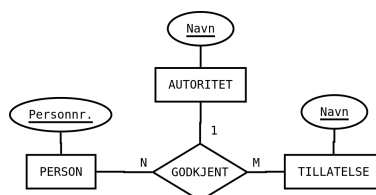


b)

Vi vil lage et ER-diagram som modellerer tillatelser. En godkjenning er noe som gis til en person av en autoritet for en gitt tillatelse. Vi har altså entitetene *person*, *autoritet* og *tillatelse*, og en relasjon *godkjenning* mellom disse. En person kan få mange tillatelser fra en autoritet; en autoritet kan gi mange personer samme tillatelse; men kun én autoritet kan gi en bestemt tillatelse til en bestemt person. En person identifiseres med et personnummer, en autoritet identifiseres med et navn og en tillatelse identifiseres med et navn.

Lag ER-modellen som fanger informasjonen over.

## Løsningsforslag



## Oppgave 2 – Realisering

Realiser ER-diagrammet laget i oppgave 1a) til et databaseskjema. Forklar hvilke valg du gjør underveis.

## Løsningsforslag

1. Starter med å lage en relasjon per entitet (hvor vi ser bort ifra multi-verdi attributtet **Mail**) og må velge en av nøklene som primærnøkkel (understreket):

**Bruker**(nummer, brukernavn, navn)

**Emne**(emnenummer, fagkode, emnenavn)

2. Så vil vi realisere svake entiteter. Her har vi kun **Resultat**. For denne må vi i tillegg til attributtene **Semester** og **Karakter** ha nøkkelattributtene til dens eiende entiteter, som da er **EMNE** og **BRUKER**. Nøkkelen blir så kombinasjonen av dens svake nøkkel og nøklene til dens eiende entiteter, altså får vi:

**Resultat**(nummer, emnenummer, fagkode, semester, karakter)

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

3. Deretter må vi realisere de vanlige relasjonene. Vi starter med **REGISTRERT\_I**, som er en M-N-relasjon, så må derfor få sitt egen tabell:

**Registrert\_i**(nummer, emnenummer, fagkode)

Videre 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 **Bruker**-relasjonen. Dersom det er vanlig at en bruker er gruppelærer vil det gi mening å legge denne inn i **Bruker**-relasjonen, men dersom det er få gruppelærere per bruker kan det gi mer mening å ha den i en egen tabell (for å unngå mange NULL-verdier i **Bruker**-relasjonen). Vi antar at det siste er sant, så lager derfor en egen tabell:

**Gruppelærer\_i**(nummer, emnenummer, fagkode)

Til lutt har vi **FORELESER\_I**-relasjonen. Denne er 1-1. Siden hvert emne må ha en foreleser, gir det mest mening å legge denne relasjonen inn som en attributt i **Emne**-relasjonen. Vi oppdaterer derfor denne tabellen slik:

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

hvor **foreleser** refererer til **Bruker**(nummer).

4. 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)`

Hele skjemaet blir da:

`Bruker(nummer, brukernavn, navn)`  
`Emne(emnenummer, fagkode, emnenavn, foreleser)`  
`Resultat(nummer, emnenummer, fagkode, semester, karakter)`  
`Registrert_i(nummer, emnenummer, fagkode)`  
`Gruppelærer_i(nummer, emnenummer, fagkode)`  
`Mail(nummer, mail)`

hvor vi har fremmednøklerne:

`Resultat(nummer) → Bruker(nummer)`  
`Resultat(emnenummer, fagkode) → Emne(emnenummer, fagkode)`  
`Emne(foreleser) → Bruker(nummer)`  
`Registrert_i(emnenummer, fagkode) → Emne(emnenummer, fagkode)`  
`Gruppelærer_i(emnenummer, fagkode) → Emne(emnenummer, fagkode)`  
`Registrert_i(nummer) → Bruker(nummer)`  
`Gruppelærer_i(nummer) → Bruker(nummer)`  
`Mail(nummer) → Bruker(nummer)`

### Oppgave 3 – Relasjonsmodellen

Gitt følgende relasjoner:

`Bok(Navn, ISBN, Serie, Forlag)`  
`Serie(SID, SerieNavn)`  
`Forlag(FID, ForlagsNavn, GrunnlagtÅr)`

hvor understrekede kolonner markerer nøkler/unike verdier (f.eks. er det ingen forlag som har samme kombinasjon av navn og året det er grunnlagt). Kolonnen `Bok(Serie)` refererer til `Serie(SID)` og `Bok(Forlag)` refererer til `Forlag(FID)`.

1. Hvilke kandidatnøkler har tabellene
2. Hvilke supernøkler har tabellene

3. Skriv et uttrykk i relasjonell algebra som finner året forlaget til boken med tittel 'Skogen' ble grunnlagt
4. Skriv et uttrykk i relasjonell algebra som finner navn på serier som inneholder en bok utgitt av et forlag grunnlagt etter år 1990

### Løsningsforslag

1. *Bok* har kandidatnøkkelen {ISBN}. *Serie* har kandidatnøkkelen {SID}. *Forlag* har kandidatnøkklene {FID} og {ForlagsNavn, GrunnlagtÅr}.
2. En supernøkkel er bare en mengde kolonner som inneholder en kandidatnøkkel, så for *Bok* har vi f.eks. {ISBN}, {Navn, ISBN, Serie}, {ISBN, Forlag, Navn}, osv. (totalt 8 forskjellige). *Serie* har supernøkklene {SID} og {SID, SerieNavn}. *Forlag* har supernøkklene {FID}, {ForlagsNavn, GrunnlagtÅr}, {FID, ForlagsNavn}, {FID, GrunnlagtÅr} og {FID, ForlagsNavn, GrunnlagtÅr}.
3.  $\pi_{\text{GrunnlagtÅr}}(\sigma_{\text{Navn}='Skogen'}(\text{Bok} \bowtie_{\text{Forlag}=\text{FID}} \text{Forlag}))$   
eller  $\pi_{\text{GrunnlagtÅr}}(\sigma_{\text{Navn}='Skogen'}(\text{Bok}) \bowtie_{\text{Forlag}=\text{FID}} \text{Forlag})$
4.  $\pi_{\text{SerieNavn}}(\text{Serie} \bowtie_{\text{SID}=\text{Serie}} \text{Bok} \bowtie_{\text{Forlag}=\text{FID}} \sigma_{\text{GrunnlagtÅr} \geq 1990}(\text{Forlag}))$   
eller  $\pi_{\text{SerieNavn}}(\sigma_{\text{GrunnlagtÅr} \geq 1990}(\text{Serie} \bowtie_{\text{SID}=\text{Serie}} \text{Bok} \bowtie_{\text{Forlag}=\text{FID}} \text{Forlag}))$

### Oppgave 4 – Normalformer og normalisering

Gitt følgende tabell:

Magasin(Navn, SerieId, SerieNavn, Forlag, Nummer, Utgitt)

hvor vi har følgende følgende FDer:

Forlag, SerieId, Nummer  $\rightarrow$  Navn

SerieId  $\rightarrow$  SerieNavn

SerieId, Nummer  $\rightarrow$  Utgitt

1. Hvilke kandidatnøkler har vi?
2. Hvilken normalform er tabellen på?
3. Dekomponer tabellen *Magasin* tapsfritt til BCNF

## Løsningsforslag

1.

Kun på venstresiden:  $\{\text{Forlag, SerieId, Nummer}\}$ . Kun på høyresiden  $\{\text{Navn, SerieNavn, Utgitt}\}$ .  
Altså må hver kandidatnøkkel inneholde  $\{\text{Forlag, SerieId, Nummer}\}$ , men aldri noen av  $\{\text{Navn, SerieNavn, Utgitt}\}$ .  $\{\text{Forlag, SerieId, Nummer}\}^+ = \{\text{Forlag, SerieId, Nummer, Navn, SerieNavn, Utgitt}\}$ , så kandidatnøkkel. Siden minimal, ingen flere kandidatnøkler.

2.

1. Starter med  $\text{Forlag, SerieId, Nummer} \rightarrow \text{Navn}$ : Venstresiden er en supernøkkel, så [BCNF].
2. Sjekker så  $\text{SerieId} \rightarrow \text{SerieNavn}$ : Venstresiden ikke supernøkkel, så ikke BCNF. Høyreside ikke nøkkelattributt, så ikke 3NF. Høyresiden er en del av en kandidatnøkkel, så ikke 2NF. Altså [1NF].
3. (Kan egentlig stoppe her!) Sjekker  $\text{SerieId, Nummer} \rightarrow \text{Utgitt}$ : Venstresiden ikke supernøkkel, høyresiden ikke nøkkelattributt, venstresiden er del av kandidatnøkkel, så [1NF].

Laveste vi fant var 1NF, så tabellen er på 1NF.

3.

1. Starter f.eks. med  $\text{SerieId} \rightarrow \text{SerieNavn}$  som bryter BCNF:  $\{\text{SerieId}\}^+ = \{\text{SerieId, SerieNavn}\}$ , så vi får to tabeller  $S_1(\text{SerieId, SerieNavn})$  og  $S_2(\text{SerieId, Navn, Forlag, Nummer, Utgitt})$ .  $S_1$  har ingen FDer som bryter BCNF.
2.  $S_2$  har FDene

$\text{Forlag, SerieId, Nummer} \rightarrow \text{Navn}$

$\text{SerieId, Nummer} \rightarrow \text{Utgitt}$

og får da kandidatnøkkel  $\{\text{Forlag, SerieId, Nummer}\}$ . Dermed bryter  $\text{SerieId, Nummer} \rightarrow \text{Utgitt}$  BCNF.  $\{\text{SerieId, Nummer}\}^+ = \{\text{SerieId, Nummer, Utgitt}\}$ , så får da tabellene  $S_{21}(\text{SerieId, Nummer, Utgitt})$  og  $S_{22}(\text{SerieId, Nummer, Navn, Forlag})$ .

3. Tabellen  $S_{21}$  har kun FDen  $\text{SerieId, Nummer} \rightarrow \text{Utgitt}$  som tilfredstiller BCNF, mens  $S_{22}$  har kun FDen  $\text{Forlag, SerieId, Nummer} \rightarrow \text{Navn}$  som også tilfredstiller BCNF.

Det endelige dekomponerte skjemaet er:

$S_1(\text{SerieId}, \text{SerieNavn})$

$S_{21}(\text{SerieId}, \text{Nummer}, \text{Utgitt})$

$S_{22}(\text{SerieId}, \text{Nummer}, \text{Navn}, \text{Forlag})$