

# Normalisering

## Oppgave 1

Gitt en relasjon (primærnøkkel med fet skrift):

Institutt: **instituttkode**, **fradato**, instituttnavn, adresse, navn\_på\_leder, sluttdato

Relasjonen kan lagre historikken for instituttledere. Instituttkoden er unik for instituttene, f.eks. er Ifis kode INF. Fradato er tiltredelsesdatoen for en instituttleder. Instituttnavn er akkurat det: navnet på instituttet. Adresse er instituttets adresse. Navn\_på\_leder er nettopp det. Sluttdato er datoen for når en leder har sluttet som leder. Dette feltet må kunne være tomt, siden vi ikke har noen sluttdato for sittende leder.

- Identifiser FD'ene ut fra rimelige antakelser.
- Lag noen lovlige forekomster (poster/tupler) i relasjonen. Lag også et par som er lovlige i henhold til FD'ene, men som opplagt ikke bør være lovlige. Lag til slutt et tuppel eller to som er ulovlige.
- Hvilken normalform er denne relasjonen på? Grunngi svaret.
- Normaliser deg oppover til 3NF.
- Er resultatet også på BCNF?

## Oppgave 2

Gitt relasjonen:

R: **A**, B, C, D, E, F

Kandidatnøkler: A og (B, C). A er primærnøkkel.

FD'er:  $D \rightarrow E$  og  $B \rightarrow F$ .

- Hvilken normalform er relasjonen på? Grunngi svaret. Fjern evt. problemet ved å dekomponere.
- Hvilken normalform har du nå? Hvorfor er den (ikke) på 3NF? Dekomponer evt. til 3NF.

## Oppgave 3

Gitt følgende to relasjoner (primærnøkler med fet skrift):

R1: **gateadresse**, **postnummer**

R2: **postnummer**, poststed

- Lag SQL-kode for å joine (denormalisere) R1 og R2 til en relasjon R med tre attributter: **gateadresse**, **postnummer**, poststed.
- Lag noen tupler som er lovlige i R1/R2 og tilsvarende tupler i R. Lag også noen som er lovlige i R, men ikke tillatt i R1/R2.
- Hva blir problemet når man skal registrere en ny post i R?
- Hvilken normalform har R? Grunngi svaret.