Normalisering

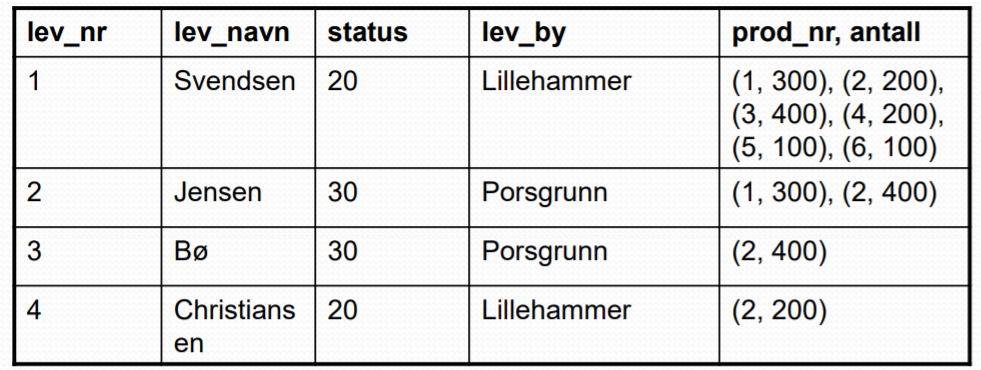
*En formell metode for å designe relasjonsdatabaser via funksjonelle avhengigheter.*

**Hensikten** med dette er at man skal unngå unødvendig dobbellagring av data (redundans) siden det lett fører til inkonsistente data. Unntaket er primærnøkkel og fremmednøkkel som brukes for å koble sammen tabeller.

Normalisering legger stor vekt på å finne kandidatnøkler.

Bruk av normalisering kan føre til kvalitetssikring eller forstå en eksisterende database.

**Tabellen som brukes som eksempel. Denne er «unormalisert», altså kan ikke brukes i database.**



# Funksjonell avhengighet

Y er *funksjonell avhengig* (f.a) av X hvis og bare hvis det til hver X-verdi er assosiert eksakt en Y-verdi. R.X -> R.Y

Y er *fullt funksjonell avhengig* (f.f.a) av X, hvis den er funksjonell avhengig av X, og det ikke fins en delmengde Z av X som Y er funksjonelt avhengig av.

# Første normalform (1NF)

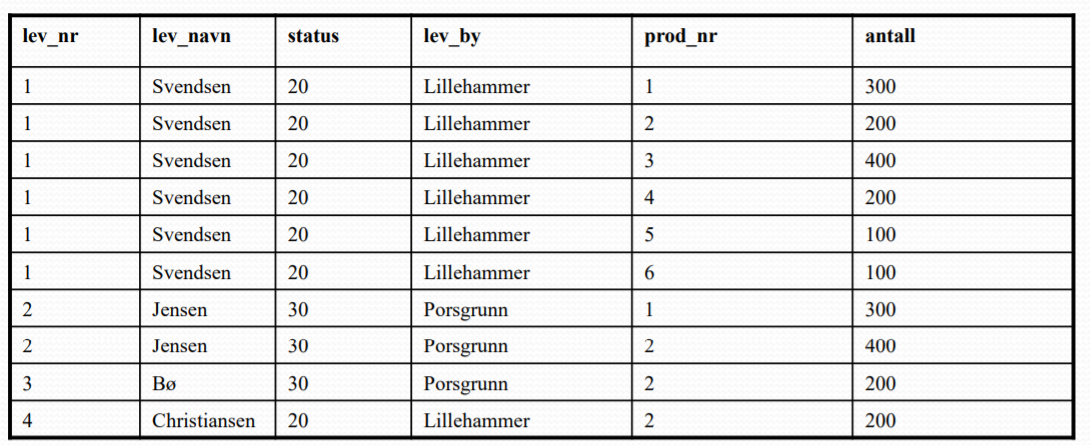
1NF: Det eksisterer en primærnøkkel, og ingen del av denne er NULL. Tabellen inneholder ikke repeterende grupper. Kun en verdi i hver rute i tabellen.

Tabellen blir da seende slik ut på 1NF:

**Primærnøkkel**: kombinasjon av lev\_nr og prod\_nr.

**Problemer med oppdatering:**

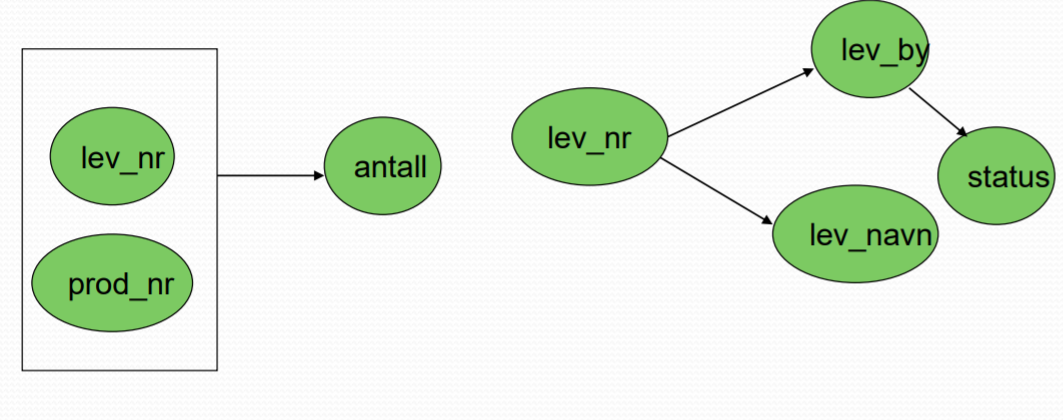
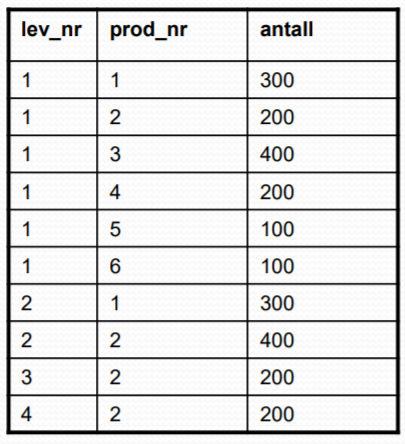
* *Insert*: Vi vil f.eks. legge inn 5, Hansen. 20, Lillehammer. Kan ikke legge inn leverandør uten at han leverer et produkt.
* *Delete*: Sletting av leveranse kan medføre at leverandørinfo slettes.
* *Update*: Endring av bynavn medfører endring for alle tilhørende leveranser.



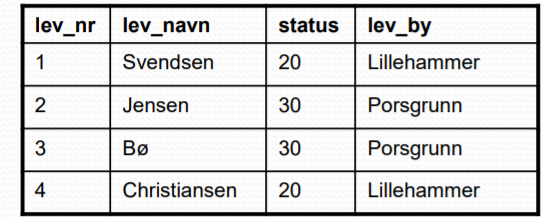
# Andre normalform (2NF)

2NF: Hvis og bare hvis en relasjon er på 1NF, og den ikke inneholder partielle avhengigheter (en ikke-nøkkel-attributt er f.a. av en del av primærnøkkelen)

Fra 1NF til 2NF: Splitt opp relasjonen i flere nye relasjoner slik at partielle avhengigheter unngås. («Ingen piler krysser bokser»).



🡨 Primærnøkkel



**Primærnøkler i tabellene ovenfor**:

* *Høyre*: lev\_nr
* *Venstre*: lev\_nr, prod\_nr

**Problemer med oppdatering venstre:**

* *Insert*: Ok
* *Delete*: Ok
* *Update*: Ok

**Problemer med oppdatering høyre:**

* *Insert*: Ok, hvis at vi vet status til byen. Får ikke til å legge til by uten leverandør.
* *Delete*: Mister byens status dersom vi sletter den siste i byen. Husk å slette tilhørende leveranser
* *Update*: Endre lev\_navn ok, endre bynavn medfører endring mange steder.

s

*Splitter ved å foreta en projeksjon. En naturlig forening bringer relasjonene tilbake til utgangspunktet.*

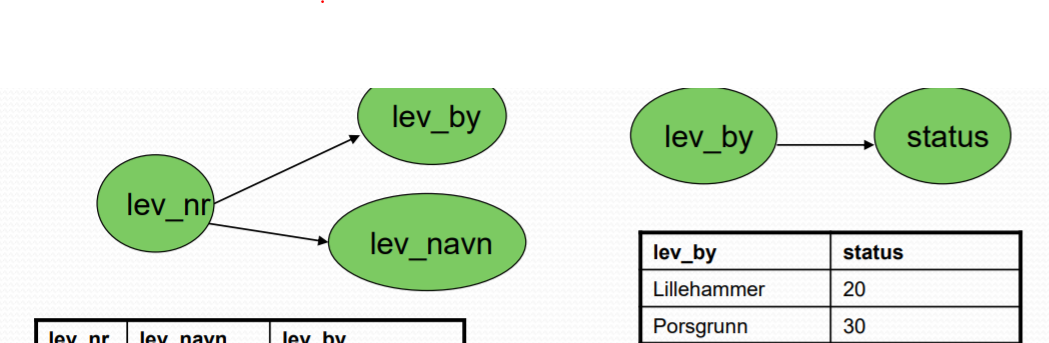
# Tredje normalform (3NF)

Fra 2NF til 3NF: *Transitive avhengigheter* fjernes. «En kjede av avhengigheter».

Tredje normalform (3NF): En relasjon er på 3NF, hvis og bare hvis den er på 2NF, og den ikke inneholder transitive avhengigheter. (”Alle piler utgår fra primærnøkkelen.”)

Transitiv avhengighet:

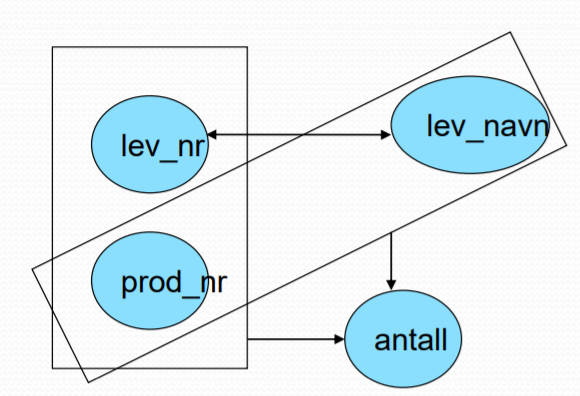
* lev\_nr 🡪 lev\_by 🡪 status, det vil si lev\_nr 🡪 status.



*Alt er ok i dette tilfellet. Både update, delete og insert.*

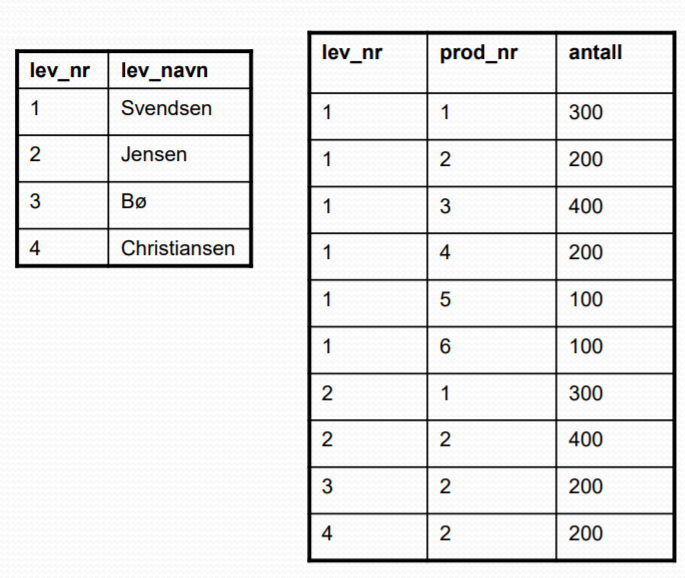
# Boyce-Codd normalform (BCNF)

BCNF: En relasjon tilfredsstiller BCNF hvis enhver determinant er en kandidatnøkkel. Hvis en tabell har bare en kandidatnøkkel, er 3NF og BCNF ekvivalente.



*I figuren er ikke BCNF oppfylt.*

**Splitter slik at BCNF er tilfredsstilt**



*3NF og BCNF forskjellig hvis:*

*1. Tabellen har to eller flere kandidatnøkler*

*2. Minst to av kandidatnøklene er sammensatte.*

*3. Minst to av kandidatnøklene har overlappende attributter.*

# Oppsummering

**Første normalform (1NF)**

* Det eksisterer en primærnøkkel, og ingen del av denne er NULL. Tabellen inneholder ikke repeterende grupper. - Kun én verdi i hver rute i tabellen.

**Andre normalform (2NF)**

* En relasjon er på 2NF hvis og bare hvis den er på 1NF, og den ikke inneholder partielle avhengigheter (en ikke-nøkkel-attributt er f.a. av en del av primærnøkkelen).

**Fra 1NF til 2NF:**

* Splitt opp relasjonen i flere nye relasjoner slik at partielle avhengigheter unngås. (”Ingen piler krysser bokser”)

**Tredje normalform (3NF)**

* En relasjon er på 3NF, hvis og bare hvis den er på 2NF, og den ikke inneholder transitive avhengigheter. (”Alle piler utgår fra primærnøkkelen.”)

**Boyce-Codd normalform (BCNF)**

* En relasjon tilfredsstiller BCNF hvis enhver determinant er en kandidatnøkkel. Hvis en tabell har bare én kandidatnøkkel, er 3NF og BCNF ekvivalente.

Mulige problemer med relasjoner som ikke tilfredsstiller 3NF/BCNF

* Oppdatering kan føre til inkonsistent data.
* Det kan være umulig å legge inn visse typer data.
* Å slette kan føre til at vi mister mer data enn det som var hensikten.

I praksis er det vanlig å normalisere en database til 3NF eller BCNF.

Noen ganger velger man lavere normaliseringsgrader, da som regel av ytelseshensyn. Dette bør i så fall være et bevisst, godt begrunnet og dokumentert valg.