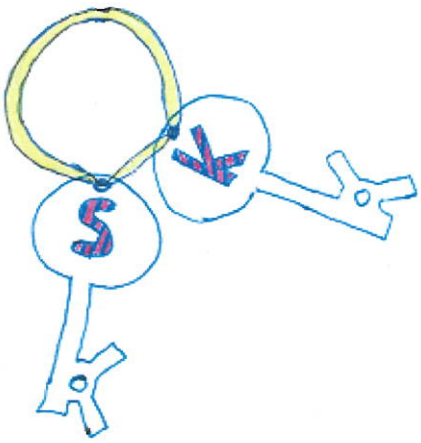
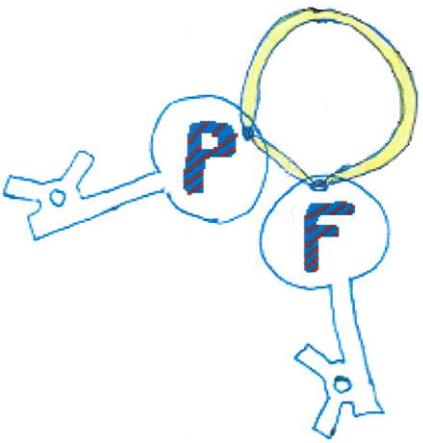




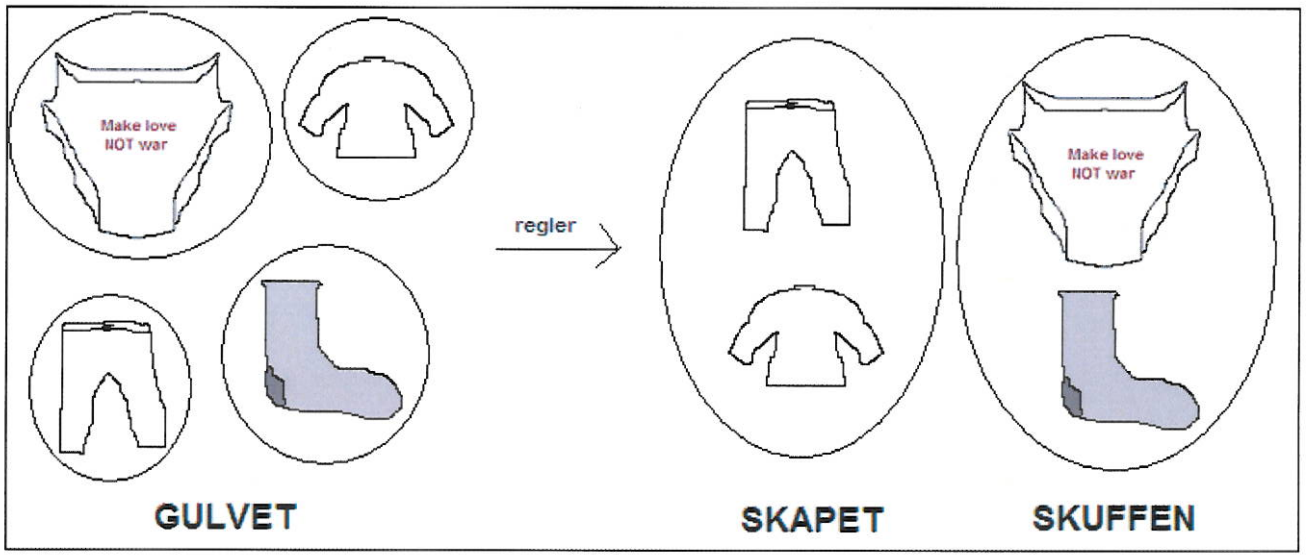
Normalisering

Elin K. Ajer Andreassen, 2009

Hva er normalisering?

Normalisering er en prosess en samling data må gjennom for å bli en god database eller en god datamodell. Med en god database menes her en samling tabeller som er godt strukturert slik at man i minst mulig grad dupliserer data.

Prosessen starter med et avgrenset, definert kaos og ender opp som en ordnet struktur.
Som en del av prosessen inngår også å sette opp hvilke forutsetninger datamodellen må forholde seg til.
For å gjennomføre normaliseringsprosessen, har man et sett med regler for hvordan man steg for steg skal komme seg fram til målet.



Hvorfor er det viktig å normalisere en datamodell eller en database?

Det er mange viktige grunner til hvorfor man må normalisere datamodeller eller databaser. Det viktigste er effektivitet. Med effektivitet menes at man må kunne oppdatere, legge inn og slette data med færrest mulig operasjoner, samtidig som mulighetene for feil i forbindelse med slike operasjoner blir redusert til et minimum. Det er også viktig med normalisering for å redusere størrelsen på en database(lagringsplass for data - kosteffektivitet) ved å forhindre overflødig data(duplisering) i databasen.

Før jeg går løs på selve normaliseringsprosessen, er det en del begreper det er viktig å kjenne til og forstå. Noen av disse begrepene er repetisjon fra tidligere, mens annet er helt nytt.
Jeg vil bruke følgende ikke-normaliserte tabell for å eksemplifisere noen av begrepene:

Barnehage

id	navn	adresse	postnr	poststed	telefon_nr	avd_id	avd_navn
1	Dyreskogen	Torvildtoppen 3	1752	Halden	69 188888	1	Rev
1	Dyreskogen	Torvildtoppen 3	1752	Halden	69 188888	2	Ulv
2	Blomsterenga	Ertemoen 3	1781	Halden	69 177777	1	Fjellfiol
3	Solgløttheimen	Solgløttveien 32	1767	Halden	69 178888	1	Lilla
3	Solgløttheimen	Solgløttveien 32	1767	Halden	69 178888	2	Oransj
3	Solgløttheimen	Solgløttveien 32	1767	Halden	69 178888	3	Turkis
4	Veivalget	Prærieheia	1782	Halden	69 185555	1	Høyre
4	Veivalget	Prærieheia	1782	Halden	69 185555	2	Venstre
5	Tittetua	Folkeveien 32	1792	Tistedal	69 199999	1	Tyttebær
6	Ysteråsen	Ysteråsen 543	1777	Halden	69 198888	1	Elg
6	Ysteråsen	Ysteråsen 543	1777	Halden	69 198888	2	Rev

Forutsetninger:

– flere barnehager kan ha samme navn, men ikke samme id

– flere barnehager kan ha samme adresse, men ikke samme telefonnr

– en barnehage kan ikke ha flere telefonnumre

– flere avdelinger innen samme barnehage kan ikke ha samme navn eller id, men det kan finnes avdelinger i andre barnehager med samme avdelingsnavn og/eller avdelings-id.

Noen viktige begreper om verdier og relasjoner

Atomær verdi:

Et attributt som ikke inneholder delverdier (det kan ikke deles opp noe mer).

Fra vårt eksempel: **adresse** kan i noen tilfeller ses på som en ikke-atomær verdi.

Repeterende grupper:

At det finnes mer enn en verdi i krysset mellom rad og kolonne i en tabell.

Universalrelasjon:

Det halvordnede kaos som vi skal rydde opp i gjennom normaliseringsprosessen. I universalrelasjonen er alle data samlet i en tabell, og det finnes ingen regler annet enn at man har rader og kolonner.

Noen viktige begreper om nøkler

Supernøkkel: Ett eller flere attributter som sammen identifiserer en entitet unikt.

Eksemplet vårt inneholder blant annet følgende supernøkler:
(id, navn, adresse, postnr, poststed, telefon_nr, avd_id, avd_navn)
(id, navn, telefon_nr, avd_id, avd_navn)
(id, navn, avd_id, avd_navn)
(id, navn, avd_id)
(id, telefon_nr, avd_id)
(navn, telefon_nr, avd_id)
(navn, telefon_nr, avd_navn)
(id, avd_id)
(id, avd_navn)
(telefon_nr, avd_id)
(telefon_nr, avd_navn)

Kandidatnøkkel: Minimalistisk supernøkkel.
Har kun det antall attributter som er nødvendig for å være unik.

Eksemplet vårt inneholder følgende kandidatnøkler:
(id, avd_id)
(id, avd_navn)
(telefon_nr, avd_id)
(telefon_nr, avd_navn)

Primærnøkkel: En kandidatnøkkel som velges som *hovednøkkel* for en tabell. (dersom en tabell har flere kandidatnøkler)

Eksemplet vårt inneholder følgende primærnøkkel:
(id, avd_id)

Fremmednøkkel: Ett eller flere attributter i en tabell som *peker på* en primærnøkkel(som også er ett eller flere attributter) i en annen tabell.

Noen viktige begreper om avhengighet

Funksjonell avhengighet: Det at ett eller flere attributter entydig bestemmer verdien av ett eller flere andre attributter.

Eksemplet vårt inneholder følgende funksjonelle avhengigheter:
(navn, adresse, postnr, poststed, telefon_nr, avd_id, avd_navn) er funksjonelt avhengig av id
(navn, adresse, postnr, poststed, telefon_nr) er funksjonelt avhengig av id
navn er funksjonelt avhengig av id
adresse er funksjonelt avhengig av id
telefon_nr er funksjonelt avhengig av id
postnr er funksjonelt avhengig av id
poststed er funksjonelt avhengig av id
poststed er funksjonelt avhengig av postnr
avd_navn er funksjonelt avhengig av (id, avd_id)

Full funksjonell avhengighet: Det at ett eller flere attributter er avhengig av hele primærnøkkelen.

Eksemplet vårt inneholder følgende fulle funksjonelle avhengigheter:
avd_navn er fullt funksjonelt avhengig av (id, avd_id)

Determinant/Determinering: Attributt eller gruppe attributter som et annet attributt(eller en gruppe attributter) er fullt funksjonelt avhengig av.

Determinering uttrykkes med tegnet $\longrightarrow$

Eksemplet vårt inneholder følgende determineringer:
id $\rightarrow$ (navn, adresse, postnr, poststed, telefon_nr, avd_id, avd_navn)
id $\rightarrow$ (navn, adresse, postnr, poststed, telefon_nr)
id $\rightarrow$ navn
id $\rightarrow$ adresse
id $\rightarrow$ telefon_nr
id $\rightarrow$ postnr
id $\rightarrow$ poststed
postnr $\rightarrow$ poststed
(id, avd_id) $\rightarrow$ avd_navn
(id, avd_navn) $\rightarrow$ avd_id
(telefon_nr, avd_id) $\rightarrow$ avd_navn
(telefon_nr, avd_navn) $\rightarrow$ avd_id
telefon_nr $\rightarrow$ id

Partiell avhengighet: Det at ett eller flere attributter er avhengig av kun deler av primærnøkkelen.

Eksemplet vårt inneholder følgende partielle avhengigheter:
(navn, adresse, postnr, poststed, telefon_nr) partielt avhengig av id
navn er partielt avhengig av id
adresse er funksjonelt avhengig av id
telefon_nr er partielt avhengig av id
postnr er partielt avhengig av id
poststed er partielt avhengig av id

Transitiv avhengighet: Det at et attributt er avhengig av et annet ikke-primærnøkkel-attributt.

Eksemplet vårt inneholder følgende partielle avhengigheter:
poststed er transitivt avhengig av postnr (skrives: id $\rightarrow$ postnr $\rightarrow$ poststed)

Fra Universalrelasjon til Boyce-Codd: Normalisering i praksis

Eksempel 1: Legekontoret

Forutsetninger:
- En lege kan bare være tilknyttet et legekontor av gangen
- En leges spesialfelt følger legen, ikke legekontor_tilknytningen
- Det kan ligge flere legekontor på samme adresse
- Alle leger på et legekontor har felles telefonnummer (sentralborddame)
- Det kan finnes flere legekontor med samme navn

Universalrelasjon

Primærnøkkel

Legekontor

lege_id	legenavn	lege_spesialfelt	tilknyttet_legekontor_id	legekontor	legekontor_adr	legekontor_tlf	legekontor_periode
1	Odvar Kniven	kirurgi	1	Ulvåsveien legekontor	Ulvåsveien 79, 1765 Halden	69 218743	1.1.2009 – 31.12.2012
			4	Sentrum legekontor	Storgata 543, 1751 Halden	69 213232	1.1.2006 – 31.12.2008
2	Solveig Skrue	gynekologi, sexologi, eurologi	2	Høyresvingen legekontor	Sommeråsen 4, 1792 Tistedal	69 215454	1.1.2006 – 31.12.2010
			3	Iddefjorden naturlegesenter	Bakke, 1765 Halden	69 184564	1.1.2002 – 31.12.2005
3	Sune Skogmann	nevrologi	2	Høyresvingen legekontor	Sommeråsen 4, 1792 Tistedal	69 215454	1.1.2005 – 31.12.2008
4	Pingelius Pang	indremedisin	3	Iddefjorden naturlegesenter	Bakke, 1765 Halden	69 184564	1.1.2008 – 31.12.2011

Definisjon av 1. normalform:
"For at en tabell skal være i 1. normalform(1NF), må den ikke inneholde repeterende grupper eller ikke-atomære verdier"

Funn som bryter mot 1. normalform:
Repeterende grupper: tilknyttet_legekontor_id, legekontor, legekontor_adresse, legekontor_tlf, legekontor_periode
Ikke-atomære verdier: lege_spesialfelt, legekontor_adr, legekontor_periode
det er et spørsmål om også legenavn er en ikke-atomær verdi, men dette avhenger av hva som er mest praktisk for den enkelte database eller datamodell

Hva må vi gjøre for å få universalrelasjonen over på 1NF?

Det er to måter å gjøre det på:
Metode 1: Flate ut tabellen ved å splitte alle rader som inneholder repeterende verdier
Metode 2: Opprette en ny tabell som inneholder de repeterende verdiene. Primærnøkkelen i den nye tabellen, blir fremmednøkkel i den gamle tabellen

Vi holder oss til metode 1!

1. Normalform

Løsning 1

lege_id	legenavn	lege_spesialfelt	tilknyttet_legekontor_id	legekontor	legekontor_adr	legekontor_postnr	legekontor_poststed	legekontor_tlf	legekontor_periode_fra	legekontor_periode_til
1	Odvar Kniven	kirurgi	1	Ulvåsveien legekontor	Ulvasveien 79	1765	Halden	69 218743	01.01.2009	31.12.2013
1	Odvar Kniven	kirurgi	4	Sentrum legekontor	Storgata 543	1751	Halden	69 213232	01.01.2006	31.12.2008
2	Solveig Skrue	gynekolog	2	Høyresvingen legekontor	Sommeråsen 4	1792	Tistedal	69 215454	01.01.2006	31.12.2010
2	Solveig Skrue	sexologi	2	Høyresvingen legekontor	Sommeråsen 4	1792	Tistedal	69 215454	01.01.2006	31.12.2010
2	Solveig Skrue	eurologi	2	Høyresvingen legekontor	Sommeråsen 4	1792	Tistedal	69 215454	01.01.2006	31.12.2010
2	Solveig Skrue	gynekologi	3	Iddefjorden naturlegesenter	Bakke	1765	Halden	69 184564	01.01.2002	31.12.2005
2	Solveig Skrue	sexologi	3	Iddefjorden naturlegesenter	Bakke	1765	Halden	69 184564	01.01.2002	31.12.2005
2	Solveig Skrue	eurologi	3	Iddefjorden naturlegesenter	Bakke	1765	Halden	69 184564	01.01.2002	31.12.2005
3	Sune Skogmann	nevrologi	2	Høyresvingen legekontor	Sommeråsen 4	1792	Tistedal	69 215454	01.01.2005	31.12.2008
4	Pingelius Pang	indremedisin	3	Iddefjorden naturlegesenter	Bakke	1765	Halden	69 184564	01.01.2008	31.12.2011

Løsning 2

lege_id	legenavn	lege_spesialfelt	lege_spesialfelt2	lege_spesialfelt3	tilknyttet_legekontor_id	legekontor	legekontor_adr	legekontor_postnr	legekontor_poststed	legekontor_tlf	legekontor_periode_fra	legekontor_periode_til
1	Odvar Kniven	kirurgi	NULL	NULL	1	Ulvåsveien legekontor	Ulvasveien 79	1765	Halden	69 218743	01.01.2009	31.12.2013
1	Odvar Kniven	kirurgi	NULL	NULL	4	Sentrum legekontor	Storgata 543	1751	Halden	69 213232	01.01.2006	31.12.2008
2	Solveig Skrue	gynekologi	sexologi	eurologi	2	Høyresvingen legekontor	Sommeråsen 4	1792	Tistedal	69 215454	01.01.2006	31.12.2010
2	Solveig Skrue	gynekologi	sexologi	eurologi	3	Iddefjorden naturlegesenter	Bakke, 1765 Halden	1765	Halden	69 184564	01.01.2002	31.12.2005
3	Sune Skogmann	nevrologi	NULL	NULL	2	Høyresvingen legekontor	Sommeråsen 4	1792	Tistedal	69 215454	01.01.2005	31.12.2008
4	Pingelius Pang	indremedisin	NULL	NULL	3	Iddefjorden naturlegesenter	Bakke	1765	Halden	69 184564	01.01.2008	31.12.2011

Hvorfor er løsning 2 en dårlig løsning?
Stikkord her er fleksibilitet og redundans.
- Det blir mange tomme felter for leger med færre enn tre spesialfelt
- Hva skjer med denne strukturen dersom en lege har mer enn tre spesialfelt?

Vi holder oss til løsning 1!

Definisjon av 2. normalform:
"For at en tabell skal være i 2. normalform(2NF), må alle attributter være fullt funksjonelt avhengig av primærnøkkelen"

Funn som bryter mot 2. normalform:
Partielle avhengigheter(p.a.a.):
legenavn p.a.a. lege_id
lege_spesialfelt p.a.a. lege_id
legekontor_tlf p.a.a. tilknyttet_legekontor_id
(legekontor, legekontor_adr, legekontor_postnr, legekontor_poststed, legekontor_tlf) p.a.a. tilknyttet_legekontor_id

Hvorfor er ikke legekontor_adr partielt avhengig av legekontor_id, mens legekontor_tlf er det?
Det kan finnes flere legekontor på samme adresse, men ikke flere legekontor med samme telefonnr.

Hva må vi gjøre for å få tabellen over på 2NF?

Vi må splitte tabellen i flere tabeller, slik at det ikke er noen partielle avhengigheter igjen

2. Normalform

Legekontor

legekontor_id	legekontor	legekontor_adr	legekontor_postnr	legekontor_poststed	legekontor_tlf
1	Ulvåsveien legekontor	Ulvåsveien 79	1765	Halden	69 218743
4	Sentrum legekontor	Storgata 543	1751	Halden	69 213232
2	Høyresvingen legekontor	Sommeråsen 4	1792	Tistedal	69 215454
3	Iddefjorden naturlegesenter	Bakke	1765	Halden	69 184564

Supernøkler: (legekontor_id, legekontor, legekontor_adr, legekontor_postnr, legekontor_poststed, legekontor_tlf)
(legekontor_id, legekontor, legekontor_adr, legekontor_postnr, legekontor_poststed)
(legekontor_id, legekontor, legekontor_adr, legekontor_postnr)
(legekontor_id, legekontor, legekontor_adr)
(legekontor_id, legekontor)
legekontor_id

Kandidatnøkler: legekontor_id

Primærnøkkel: legekontor_id

Lege

lege_id	legenavn
1	Odvar Kniven
2	Solveig Skrue
3	Sune Skogmann
4	Pingelius Pang

Supernøkler: (lege_id, legenavn)
lege_id

Kandidatnøkler: lege_id

Primærnøkkel: lege_id

LegeSpesialfelt

lege_id	lege_spesialfelt
1	kirurgi
2	gynekolog
2	sexologi
2	eurologi
3	nevrologi
4	indremedisin

Supernøkler: (lege_id, lege_spesialfelt)

Kandidatnøkler: (lege_id, lege_spesialfelt)

Primærnøkkel: (lege_id, lege_spesialfelt)

LegeLegekontor

lege_id	legekontor_periode_fra	legekontor_periode_til	tilknyttet_legekontor
1	01.01.2009	31.12.2013	1
1	01.01.2006	31.12.2008	4
2	01.01.2006	31.12.2010	2
2	01.01.2002	31.12.2005	3
3	01.01.2005	31.12.2008	2
4	01.01.2008	31.12.2011	3

Supernøkler: (lege_id, legekontor_periode_fra, legekontor_periode_til, tilknyttet_legekontor)
(lege_id, legekontor_periode_fra, legekontor_periode_til)
(lege_id, legekontor_periode_fra)
(lege_id, legekontor_periode_til)

Kandidatnøkler: (lege_id, legekontor_periode_fra)
(lege_id, legekontor_periode_til)

Primærnøkkel: (lege_id, legekontor_periode_fra) *Fordi må velge en av kandidatnøklene*

Definisjon av 3. normalform:
"For at en tabell skal være i 3. normalform(3NF), må det ikke forekomme noen transitive avhengigheter"

Determineringer/Funksjonell avhengighet:

Legekontor legekontor_id → legekontor
legekontor_id → legekontor_tlf
legekontor_postnr → legekontor_poststed

Lege lege_id → legenavn

LegeSpesialfelt (lege_id, lege_spesialfelt) → (lege_id, lege_spesialfelt)

LegeLegekontor (lege_id, legekontor_periode_fra) → legekontor_periode_til
(lege_id, legekontor_periode_til) → legekontor_periode_fra

(lege_id, legekontor_periode_fra) → tilknyttet_legekontor
(lege_id, legekontor_periode_til) → tilknyttet_legekontor

Funn som bryter mot 3. normalform:
Transitive avhengigheter(t.a.a.):
legekontor_poststed t.a.a. legekontor_postnr

Hva må vi gjøre for å få tabellen over på 3NF?

Vi må splitte tabellen i flere tabeller, slik at det ikke er noen transitive avhengigheter igjen

3. Normalform

Legekontor

legekontor_id	legekontor	legekontor_adr	legekontor_postnr	legekontor_tlf
1	Ulvåsveien legekontor	Ulvåsveien 79	1765	69 218743
4	Sentrum legekontor	Storgata 543	1751	69 213232
2	Høyresvingen legekontor	Sommeråsen 4	1792	69 215454
3	Iddefjorden naturlegesenter	Bakke	1765	69 184564

Determinanter: legekontor_id legekontor_id → (legekontor, legekontor_adr, legekontor_postnr, legekontor_tlf)

Kandidatnøkler: legekontor_id

Fremmednøkler: postnr

Poststed

postnr	poststed
1765	Halden
1751	Halden
1792	Tistedal

Determinanter: postnr postnr → poststed

Kandidatnøkler: postnr

Fremmednøkler: ingen

Lege

lege_id	legenavn
1	Odvar Kniven
2	Solveig Skrue
3	Sune Skogmann
4	Pingelius Pang

Determinanter: lege_id lege_id → legenavn

Kandidatnøkler: lege_id

Fremmednøkler: ingen

LegeSpesialfelt

lege_id	lege_spesialfelt
1	kirurgi
2	gynekolog
2	sexologi
2	eurologi
3	nevrologi
4	indremedisin

Determinanter: (lege_id, lege_spesialfelt) (lege_id, lege_spesialfelt) → (lege_id, lege_spesialfelt)

Kandidatnøkler: (lege_id, lege_spesialfelt)

Fremmednøkler: lege_id

LegeLegekontor

lege_id	legekontor_periode_fra	legekontor_periode_til	tilknyttet_legekontor
1	01.01.2009	31.12.2013	1
1	01.01.2006	31.12.2008	4
2	01.01.2006	31.12.2010	2
2	01.01.2002	31.12.2005	3
3	01.01.2005	31.12.2008	2
4	01.01.2008	31.12.2011	3

Determinanter: (lege_id, legekontor_periode_fra) (lege_id, legekontor_periode_fra) → legekontor_periode_til og (lege_id, legekontor_periode_fra) → tilknyttet_legekontor
(lege_id, legekontor_periode_til) (lege_id, legekontor_periode_til) → legekontor_periode_til og (lege_id, legekontor_periode_til) → tilknyttet_legekontor

Kandidatnøkler: (lege_id, legekontor_periode_fra)
(lege_id, legekontor_periode_til)

Fremmednøkler: lege_id
tilknyttet_legekontor

Definisjon av Boyce-Code normalform:
"For at en tabell skal være i Boyce-Code normalform(BCNF), må alle determinanter være kandidatnøkler "

Vi har ingen brudd på BCNF

Det er ytterst sjelden at det er brudd på BCNF, men det er viktig å sjekke det likevel!

Her er et par tilfeller som KAN føre til brudd på BCNF:
- En relasjon som inneholder to eller flere kombinerte kandidatnøkler
- Kandidatnøklene har minst ett felles attributt

Litt SQL

Ved hjelp av SQL kan man lage en spørring som viser data i 1. normalform:

```
SELECT l.lege_id, l.legenavn, lsf.lege_spesialfelt, llk.tilknyttet_legekontor, lk.legekontor, lk.legekontor_adr, lk.legekontor_postnr, p.poststed AS legekontor_poststed, lk.legekontor_tlf, llk.legekontor_periode_fra, llk.legekontor_periode_til
FROM Legekontor lk, Lege l, LegeSpesialFelt lsf, LegeLegekontor llk, Poststed p
WHERE l.lege_id = llk.lege_id
AND l.lege_id = lsf.lege_id
AND lk.legekontor_id = llk.tilknyttet_legekontor
AND lk.legekontor_postnr = p.postnr;
```

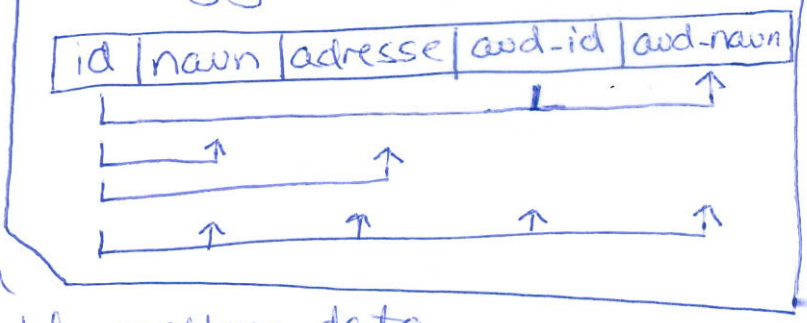
Noen siste ord

Det er ikke så ofte man behøver å være så omstendelig i sin normaliseringsprosess.
Ofte ser man tidlig strukturen for databasen eller datamodellen, og kan hoppe rett fra 1. normalform til 3. normalform.
Den omstendelige versjonen er viktig for å forstå prosessen, og for å bli fortrolig med normaliseringens verktøykasse.
Det er uansett viktig å sjekke sluttproduktet sitt for eventuelle brudd på de forskjellige normalformene.

OPPSUMMERING

Mål:
Lage DB

En måte å tegne funksjonelle avhengigheter



Lage en nøyaktig representasjon av:
- data
- forhold mellom data
- krav / begrensninger til data

Modellering
↓
Datastruktur
↓
Normalisering
(kontroll)

Normalisering
(struktur)
↓
Datastruktur

En god relasjon =

* minimalt antall attributter nødvendig for å oppfylle krav til data i systemet.
* attributter m/nær relasjon (logisk sett) ligger i samme tabell
* minimalt med redundans. helst kun fremmednøkler (for sammenkobling) > 1 gang.

Denormalisering: "Kontrollert redundans".
Kan i noen tilfeller forsvares av effektivitetshensyn.

Det finnes flere normalformer etter BCNF, men vi stopper på 3NF eller BCNF.

Bibliografi

Connolly Thomas, and Carolyn Begg. Database Systems. A Practical Approach to Design, Implementation, and Management. 4th ed. Harlow: Addison-Wesley, 2005.

Bostrøm, Edgar. Databaser – noen temaer. MetodeData a.s., 2008.