

Normalisering af en database. Hvad er det?

Hvis man prøver at sætte ord på hvad **normalisering af databaser** betyder, så er det, at man foretager en kvalitetssikring af databasen. Det vil sige, at man sørger for at der ikke opstår redundans og nulls i sine tabeller, og er med til at gøre databasen en anelse hurtigere, og meget lettere at vedligeholde. Man taler om godt database-design når man har normaliseret sin database.

Der findes fem normalformer, samt en reformulering af den tredje. Denne kaldes for Boyce/Codd normalformen.

En god ting ved **normalisering** er, at lige meget hvilken normalform man vil normalisere sine databaser på, så bruger man samme metode til det.

Denne metode går ud på, at hvis man har en tvivlsom relation, så deler man denne op i to, uden at miste data vel at mærke.

Her følger en gennemgang af de første tre normalformer

Første normalform

- Første normalform, forkortes ofte som 1. NF.

Definition: En relation er på første normalform, hvis ingen af dens domæner har elementer, der i sig selv er mængder.

- De enkelte felter må kun indeholde én værdi.
- Der må ikke være kolonner som gentager sig
- Hver entitet skal have en primær nøgle.

Hvis vi kigger på den tabel (BIBLIOTEK) nedenfor, kan vi se at der er lavet et system over et bibliotek, hvori man kan se forskellige oplysninger på et udlån. bl.a. hvem der har lånt, hvad personen har lånt, data om bogen og en dato for udlånet.

BIBLIOTEK:

laanerId	Navn	Gade	Postnr	Bynavn	bogId1	Forfatter1	Titel1	Forlag1	bogId2	Forfatter2	Titel2	Forlag2	dato
101	Poul	Avej	8260	Viby J	1	Klaus R	Heste	Dyreforlag	2	Bente F	Grise	Dyreforlag	1/1-08
101	Poul	Avej	8260	Viby J	4	Jørgen P	Både	Skibsforlag	<null>	<null>	<null>	<null>	1/1-08
201	Britt	Bvej	8000	Århus	3	Bente F	Får	Dyreforlag	<null>	<null>	<null>	<null>	2/1-08

Man kan hurtigt se at denne tabel ikke overholder første normalform, da der vil opstå nulls, hvis en låner kun låner bogen en gang, og der vil være mangel på felter hvis en låner, låner flere end to bøger.

Derfor bruger vi den løsning som jeg omtalte i min introduktion til normalisering. Vi deler tabellen op.

LAANER:

laanerId	Navn	Gade	Postnr	Bynavn
101	Poul	Avej	8260	Viby J
201	Britt	Bvej	8000	Århus

UDLAAN:

laanerId	bogId	Forfatter	Titel	Forlag	dato
101	1	Klaus R	Heste	Dyreforlag	1/1 – 08
101	2	Bente F	Grise	Dyreforlag	1/1 – 08
101	4	Jørgen P	Både	Skibsforlag	1/1 – 08
201	3	Bente F	Får	Dyreforlag	2/1 – 08

Nu har vi lavet to tabeller som begge overholder første normalform.

Vi har en tabel over lånerne (LAANER), så de ikke skal indtastes flere gange, men kan identificeres på deres id (laanerId).

Derudover har vi lavet en udlånstabel (UDLAAN), hvori man kan skrive alle de lån som bliver foretaget.

Så man vil ikke få for mange eller for lidt felter, og derved heller ikke NULL's.

Denne tabel vil jeg arbejde videre med i anden normalform.

Anden normalform

- Anden normalform forkortes, på samme måde som den første, bare med et 2-tal i stedet for. Nemlig 2. NF.

Definition: En relation R er på anden normalform, hvis den er på første normalform, og hvis enhver ikke-nøgle-attribut er fuldt funktionelt afhængig af enhver kandidatnøgle i R.

- Første normalform SKAL være opfyldt.

- Ingen attributter der ikke selv tilhører nøglen, må afhænge af en del af nøglen (Det opstår tit hvis man har en primær nøgle som er sammensat af to eller flere felter).

Hvis vi ser på tabellen UDLAAN nedenfor, kan vi se flere ting som ikke stemmer overens med anden normalform. Der er også ting som bare direkte er upraktiske.

UDLAAN:

laanerId	bogId	Forfatter	Titel	Forlag	dato
101	1	Klaus R	Heste	Dyreforlag	1/1 – 08
101	2	Bente F	Grise	Dyreforlag	1/1 – 08
101	4	Jørgen P	Både	Skibsforlag	1/1 – 08
201	3	Bente F	Får	Dyreforlag	2/1 – 08

- Primær nøglen består af en sammensætning af 'laanerId' og 'bogId'
 - Oplysninger om en bog afhænger kun af 'bogId' og derved kun en del af nøglen.
- Dette kaldes også for funktionel afhængighed.

- Data om bogen blandes sammen med data om udlånet.
- En låner kan låne samme bog flere gange på én dag.

Derfor gør vi som ved første normalform, og deler relationen (læs. tabellen) op i to:

UDLAAN:

udlaansId	laanerId	bogId	dato
1	101	1	1/1 - 08
2	101	2	1/1 - 08
3	101	4	1/1 - 08
4	201	3	2/1 - 08

BOGTABEL:

bogId	Forfatter	Titel	Forlag
1	Klaus R	Heste	Dyreforlag
2	Bente F	Grise	Dyreforlag
3	Bente F	Får	Dyreforlag
4	Jørgen P	Både	Skibsforlag

Nu har vi fået to nye og forbedrede tabeller.

For det første har vi adskilt data om bøgerne og data om udlånet.

Dernæst har vi også fået delt primærnøglen op, så vi ikke længere har en sammensat nøgle.

Det kan ses i BOGTABEL, hvor de forskellige oplysninger om en given bog, nu består af hele nøglen (bogId).

Læg også mærke til at jeg har lavet en ny primær nøgle i UDLAAN i stedet for 'laanerId'.

Jeg har oprettet 'udlaansId', så hvert udlån får sit eget id. Jeg har dog stadig beholdt 'laanerId' i UDLAAN, da vi skal kunne identificere hvilken låner der er registreret til et givent udlån.

Tredje normalform

- Tredje normalform forkortes 3. NF. i lighed med første og anden normalform.

Definition: En relation R er på tredje normalform, hvis den er på anden normalform og det gælder, at ingen ikke-nøgle-attribut er transitivt afhængig af nogen kandidatnøgle i R.

- **Anden normalform SKAL være opfyldt.**
- **Ingen attributter må afhænge af andre attributter, der ikke selv er nøgler.**

Ser vi på tabellen LAANER kan vi se at der er ting som ikke stemmer overens med reglerne i tredje normalform.

LAANER:

laanerId	Navn	Gade	Postnr	Bynavn
101	Poul	Avej	8260	Viby J
201	Britt	Bvej	8000	Århus

- For det første er 'bynavn' og 'postnr' afhængige af hinanden, men ikke af 'laanerId' som er primærnøgle.

- 'navn', 'gade', 'postnr', 'bynavn' er afhængige af 'laanerId', men samtidig er 'bynavn' også afhængig af 'postnr'.

Det hedder sig at 'bynavn' er transitivt afhængig af 'laanerId'. Indirekte afhængig af den.

Det skrives således: (laanerId → navn, gade, postnr, bynavn men også postnr → bynavn)

Jeg deler derfor tabellen LAANER op i to, så vi får:

LAANER:

laanerId	Navn	Gade	Postnr
101	Poul	Avej	8260
201	Britt	Bvej	8000

POSTNUMMER:

Postnr	Bynavn
8000	Århus
8260	Viby J

(Forestil dig at tabellen POSTNR er en tabel over alle postnumre og byer i Danmark). Efter jeg har delt tabellen i to, er 'bynavn' ikke længere afhængig af en attribut (postnr), som ikke er en nøgle.

'postnr' findes stadig i tabellen LAANER, da vi skal kunne identificere hvilken by en given låner kommer fra. Derved er begge tabeller normaliseret efter tredje normalform.