# Normalisering

Formålet med normalisering, er at rettelser i databasen skal udføres med med mindst mulig indflydelse på systemet. Man vil gerne undgå at gemme den samme data flere steder, så en evt. ændring i databasen ikke kræver ændringer mange steder.

Den redundante data har en effekt på performance og maintainability, da man bruger unødvendige ressourcer og opdatering skal ske flere steder.

Der er mange normalformer, dog normaliseres der kun til og med 3. normalform. Det skyldes at 4. og 5. normalform, for at det meste er opfyldte efter de første 3, og det kan være specifikt for domænet om det er nødvendigt om de efterfølgende bør være opfyldt.

|  |  |  |
| --- | --- | --- |
| NF | Beskrivelse | Huskeregel |
| 1NF | Hvis der er repeterende felter, tages de over i deres egen tabel med foreginkey til den oprindelige tabel.  Her er det påkrævet, at der eksisterer en primærnøgle. Hvis denne nøgle har ens attributter(pris1 og pris2), så skal de kolonner fjernes fra tabellen, og lave deres egen tabel med primærnøglens id | Ingen repeterende felter |
| 2NF | 1NF skal være overholdt.  Hvis tabellen indeholder en sammensat primærnøgle, skal de alle resterende felter være funktionelt afhængige af primærnøglen.  Det vil sige, at de felter der ikke indgår i nøglen, laver en ny tabel sammen med en del fra den sammenhængende nøgle. Denne delnøgle blvier derfor primærnøglen for den nye tabel. | Alle felter skal være funktionelt afhængig af primærnøgle |
| 3NF | Skal opfylde 2. normalform.  Der må ikke findes felter uden for primærnøglen, som er indbyrdes afhængige.  Et klassisk eksempel medlemstabellen, hvor by og postnummer indgår. Da by er afhængige af postnr, men postnr ikke indgår i primærnøglen, så fjernes by feltet og laver en ny tabel med postnr som primærnøglen. | Postnummer |
| 4NF | Skal opfylde 3. normalform.  Et klassisk eksempel på 4. normalform er pizzaeksemplet. Et pizzaria leverer flere typer pizzaer, fx tyk og tynd bund. Pizzariaet lever til flere byer. Hvis pizzeriaet leverer alle typer pizza til alle områder hvor de leverer, er tabellen til venstre en fin løsning. Hvis de leverer bestemte typer pizzaer til bestemte områder skal tabellen deles op i de to mindre til højre. | Pizza |

|  |  |
| --- | --- |
| Restaurant | Levering |
| Pizzahut | Århus |
| Pizzahut | Skive |

|  |  |  |
| --- | --- | --- |
| Restaurant | Pizzabund | Levering |
| Pizzahut | Tynd bund | Århus |
| Pizzahut | Tyk bund | Århus |
| Pizzahut | Tynd bund | Skive |
| Pizzahut | Tyk bund | Skive |

|  |  |
| --- | --- |
| Restaurant | Pizzabund |
| Pizzahut | Tynd bund |
| Pizzahut | Tyk bund |

# Funktionelle afhængigheder

Funktionelle afhængigheder beskriver afhængighederne to sæt attributter i en relationel database. Dette kan bruges som værktøj til at primærnøgler og til at normalisere sin database. Et eksempel på en FD kunne være:  
  
FD1: Medarbejdernummer -> medarbejdernavn   
(medarbejdernummer udpeger et bestemt medarbejdernavn)