



NTNU

Norwegian University of
Science and Technology

TDT4145 Datamodellering og databasesystemer

Normaliseringsteori, del 1

13. februar 2017

Roger Midtstraum (roger.midtstraum@ntnu.no)

Funksjonelle avhengigheter

- $X \rightarrow Y$ der $X, Y \subseteq R$ uttrykker en **restriksjon** på alle lovlige tabellforekomster:
 - Alle tupler, t_i og t_j , i en forekomst $r(R)$
 - som har samme verdier for attributtene i X
 - $t_i[X] = t_j[X]$
 - *må* ha samme verdier for attributtene i Y
 - $t_i[Y] = t_j[Y]$
- Eksempler:
 - $\text{Regnr} \rightarrow \text{Navn}, \text{Rase}, \text{Får}, \text{EierPnr}$
 - $\text{Regnr}, \text{Pnr} \rightarrow \text{Antall}$

Utleidningsregler

IR-1 (reflexive): Hvis $Y \subseteq X$ så $X \rightarrow Y$

IR-2 (augmentation): $\{ X \rightarrow Y \}$ gir $XZ \rightarrow YZ$

IR-3 (transitive): $\{ X \rightarrow Y, Y \rightarrow Z \}$ gir $X \rightarrow Z$

IR-4 (decomposition): $\{ X \rightarrow YZ \}$ gir $X \rightarrow Y$

IR-5 (additive): $\{ X \rightarrow Y, X \rightarrow Z \}$ gir $X \rightarrow YZ$

IR-6 (pseudotransitive): $\{ X \rightarrow Y, WY \rightarrow Z \}$ gir $WX \rightarrow Z$

$X, Y, Z, W \subseteq R$ (mengden av alle attributter)

IR-1 + IR-2 + IR-3 kalles *Armstrongs aksiomer* og er tilstrekkelig for å utlede alle funksjonelle avhengigheter fra et gitt utgangspunkt (en mengde funksjonelle avhengigheter).

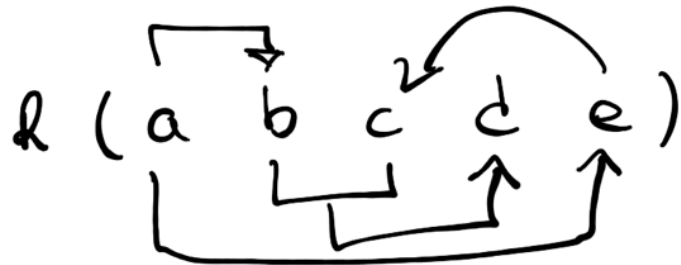
Tillukningen til en mengde FD-er

- F er en mengde funksjonelle avhengigheter
- $F^+ = \{ X \rightarrow Y \mid X \rightarrow Y \text{ kan utledes fra FD-ene i } F \}$
 - Opplisting av alle FD-er som gjelder når vi har F .
 - Merk: F og F^+ uttrykker akkurat samme restriksjon
- Eksempel:
 - Anta $R = \{a, b, c\}$ og $F = \{a \rightarrow b; b \rightarrow c\}$
 - $F^+ = \{a \rightarrow a; a \rightarrow b; a \rightarrow c; a \rightarrow ab; a \rightarrow abc; ab \rightarrow a; abc \rightarrow a; \text{ osv. } \}$
- F^+ inneholder mange *trivielle* funksjonelle avhengigheter
 - $X \rightarrow Y$, der $Y \subseteq X$. Uttrykker ingen restriksjon
- Krevende å beregne, men heldigvis sjelden interessant

Tillukningen til en mengde attributter

- Anta R og F . $X \subseteq R$
- $X^+ = \{ Y \in R \mid X \rightarrow Y \in F^+ \}$
 - Alle attributter som er funksjonelt avhengig av X
 - $X \rightarrow X^+$ vil gjelde
- Algoritme: $x^+ = x;$
repeat
 $oldX^+ = X^+;$
 for each $Y \rightarrow Z \in F$ do
 if $Y \subseteq X^+$ then
 $X^+ = X^+ \cup Z;$
until $X^+ = oldX^+;$

Eksempel



$$F = \{a \rightarrow b; b \rightarrow c; c \rightarrow d; d \rightarrow e; e \rightarrow a\}$$

$$a^+ = abecd = abcde = R$$

$$b^+ = b$$

$$d^+ = d$$

$$c^+ = c$$

$$e^+ = ec$$

$$be^+ = becd$$

NTNU

Norwegian University of
Science and Technology

Oppgave

- a) (3 %) Ta utgangspunkt i tabellen **Birds**(SpeciesID, SpeciesName, BirdGroup, Prevalence) og følgende tabellforekomst:

SpeciesID	SpeciesName	BirdGroup	Prevalence
1	Redwing	Thrushes	Migratory
2	Blackbird	Thrushes	Resident
3	Raven	Crows	Resident
4	Magpie	Crows	Resident
5	Chaffinch	Finches	Resident

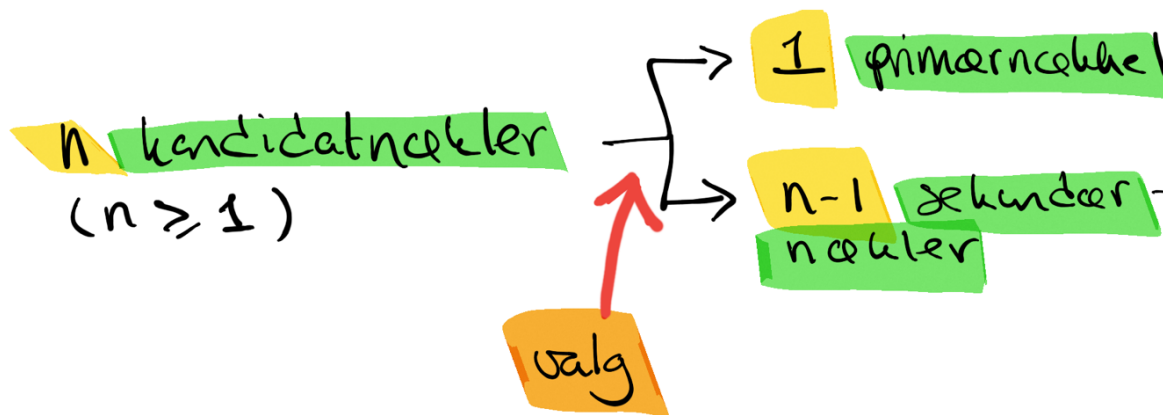
Hvilke funksjonelle avhengigheter (eng: functional dependencies) er det rimelig å anta vil gjelde for denne tabellen? Forklar de forutsetningene du legger til grunn. Du trenger ikke å ta med trivielle funksjonelle avhengigheter eller funksjonelle avhengigheter som kan utledes ut fra funksjonelle avhengigheter i svaret ditt.

Supernøkler og nøkler

- En **supernøkkel** for en tabell R er en mengde attributter S slik at:
 - Det ikke i noen forekomst av tabellen kan finnes to tupler, t_i og t_j , med samme verdier for S ($t_i[S] = t_j[S]$).
 - Supernøkkel vil være en *unik identifikator* for tabellen.
 - $S^+ = R$
- En **nøkkel** K er en *minimal* supernøkkel
 - Vi kan ikke fjerne noe attributt fra K og fortsatt ha en supernøkkel.
 - Alle nøkler er supernøkler, noen supernøkler er nøkler

Kandidat-, primær- og sekundærnøkler

- Alle tabeller vil ha minst en *nøkkel*
- En tabells nøkler utgjør tabellens *kandidatnøkler*
- En kandidatnøkkel velges til *primærnøkkel*
- Øvrige kandidatnøkler utgjør tabellens *sekundærnøkler* (alternative nøkler)



Eksempel

Student ($Pid, Sid, Navn, F\ddot{a}r$)

$F = \{Pid \rightarrow Sid; Sid \rightarrow Pid, Navn, F\ddot{a}r\}$

$Pid^+ = Pid, Sid, Navn, F\ddot{a}r$

Key Superkey

$Sid^+ = Sid, Pid, Navn, F\ddot{a}r$

K SK

$Navn^+ = Navn$

TK TSK

$F\ddot{a}r^+ = F\ddot{a}r$

TK TSK

$(Pid, Sid)^+ = Pid, Sid, Navn, F\ddot{a}r$

TK SK

$(Pid, Navn)^+ = Pid, Sid, Navn, F\ddot{a}r$

TK SK

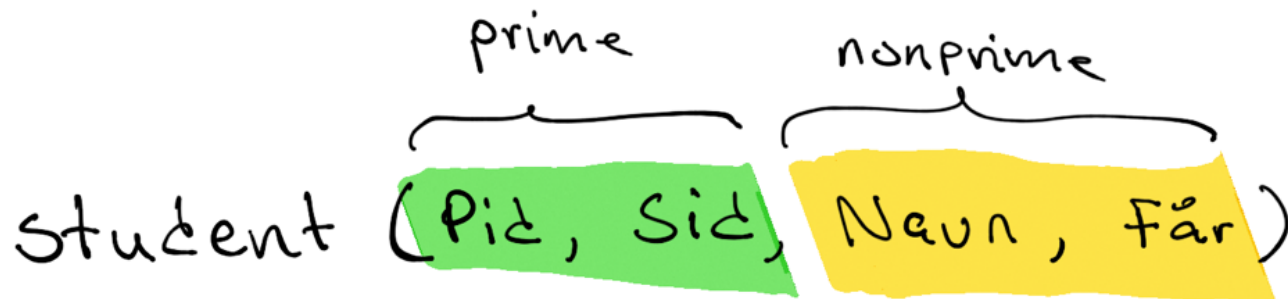
Student har 2 kandidatnøkler, mange supernøkler

Oppgave

- b) (3 %) Gitt $R = \{A, B, C, D, E\}$ og $F = \{A \rightarrow B, B \rightarrow C, CD \rightarrow E\}$. Finn alle kandidatnøkler i R . Svaret må begrunnes.

Nøkkel- og ikke-nøkkel-attributter

- *Nøkkelattributt* (eng: prime)
 - Attributter som inngår i en eller flere kandidatnøkler
- *Ikke-nøkkelattributt* (nonprime)
 - Attributter som ikke inngår i noen kandidatnøkkel
- Partisjonerer attributtene i en tabell i 2 deler



Normalformer

- Regler som stiller stadig strengere krav til tabeller
 - Sikrer at vi har tabeller som unngår uheldige egenskaper
- **Første normalform (1NF)**
 - Attributtenes domener inneholder atomiske (udelelige) verdier
 - Verdien til et attributt er en enkelt verdi fra domenet
 - Sikrer "flate, 2-dimensjonale tabeller"
 - Unngår sammensatte attributter, flere verdier og nøstede tabeller
 - NB! Det finnes tabeller som ikke er på 1NF
- Alle høyere normalformer forutsetter 1NF
- Alle tabeller på 2NF er på 1NF, alle på 3NF er på 2NF, osv.

Full funksjonell avhengighet

- En funksjonell avhengighet

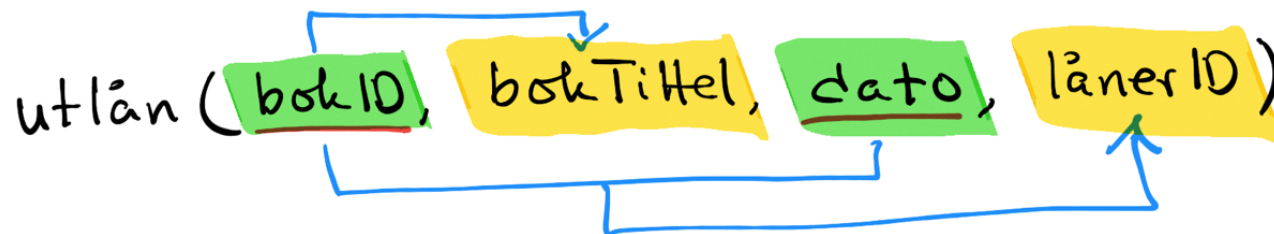
$$X \rightarrow Y$$

er en *full* funksjonell avhengighet hvis det er umulig å fjerne et attributt, $A \in X$, og ha $(X - \{A\}) \rightarrow Y$

- Inneholder ikke "overflødige" venstreside-attributter
- Kan tenke på den som en "sterkere" regel enn en *delvis* funksjonell avhengighet, der vi kan fjerne venstreside-attributt og ...
- Sid, Pid \rightarrow Navn er en delvis f.a.
- Sid \rightarrow Navn og Pid \rightarrow Navn er fulle f.a.

Andre normalform (2NF)

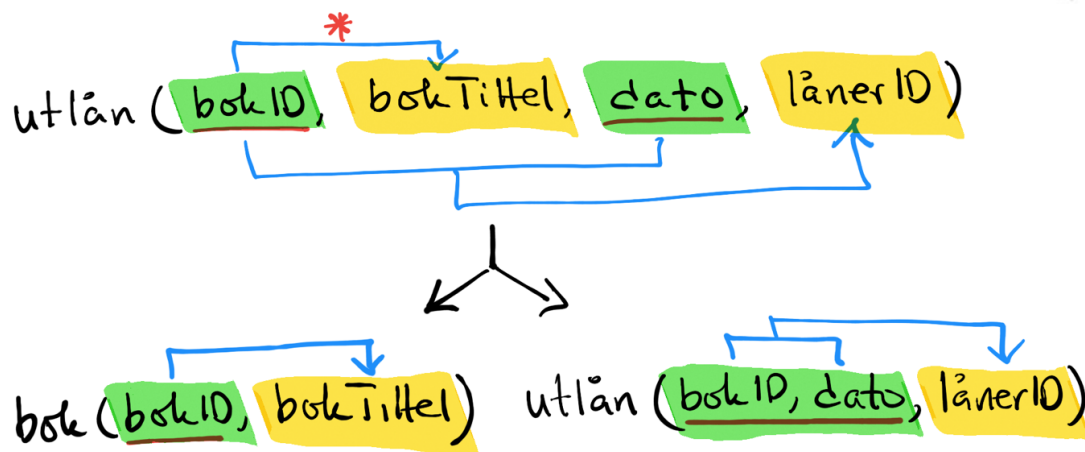
- En tabell er på andre normalform hvis og bare hvis
 - Det ikke finnes noe ikke-nøkkel-attributter som er delvis avhengig av en kandidatnøkkel



- Kandidatnøkkel: bokID, dato (maks ett utlån av samme bokID per dato)
- bokID, dato → lånerID er en full f.a.
- bokID, dato → bokTittel er en delvis f.a.
 - bokTittel er et ikke-nøkkel-attributt som er delvis avhengig av en kandidatnøkkel
 - Utlån-tabellen er ikke på 2NF

Å oppnå andre normalform

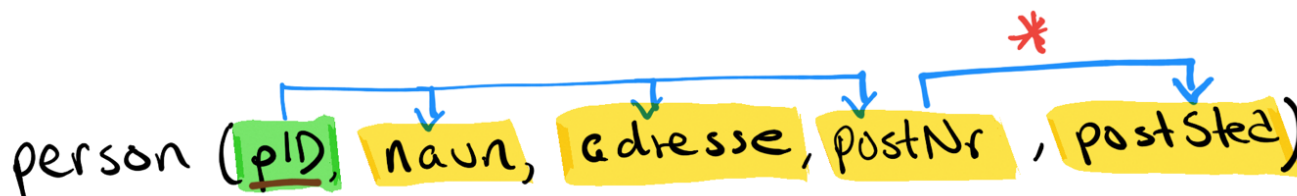
- Problem
 - bokTittel henger "tettere sammen" med bokID enn med hele kandidatnøkkelen
- Løsning
 - Splitte i to tabeller



- bok og nye utlån er begge på 2NF
- Har fjernet kilde til redundans (delvis avhengighet av nøkkel)

Tredje normalform (3NF)

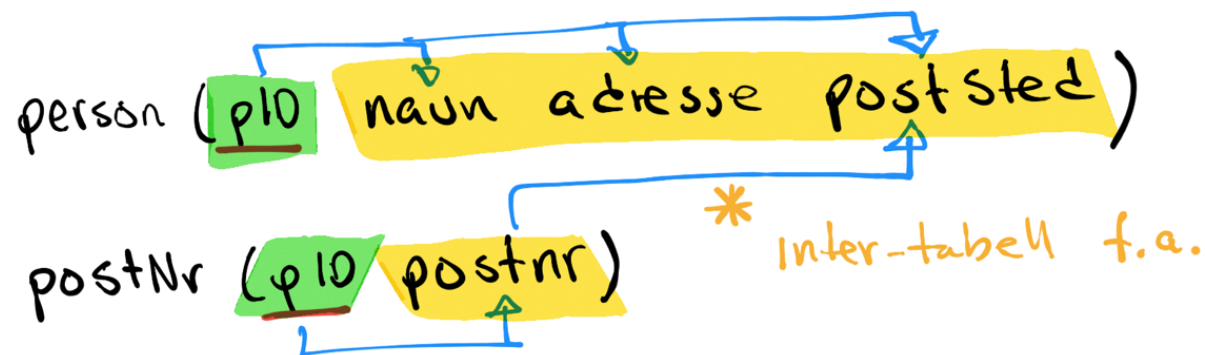
- En tabell er på **tredje normalform** hvis og bare hvis det for alle funksjonelle avhengigheter på formen, $X \rightarrow A$, som gjelder for tabellen er slik at:
 - X er en supernøkkel i tabellen, eller
 - A er et nøkkelattributt i tabellen
- Husk: $X \rightarrow BCD$ utleder $X \rightarrow B$; $X \rightarrow C$; $X \rightarrow D$



- $\text{postNr} \rightarrow \text{postSted}$
 - postNr er ikke en supernøkkel
 - postSted er ikke et nøkkelattributt
 - person-tabellen er ikke på 3NF, men den er på 2NF

Å oppnå tredje normalform (1)

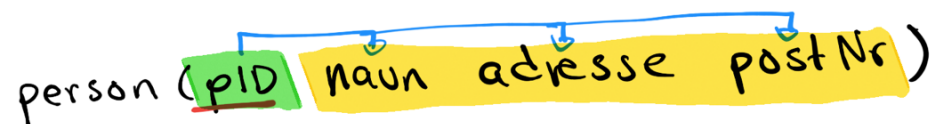
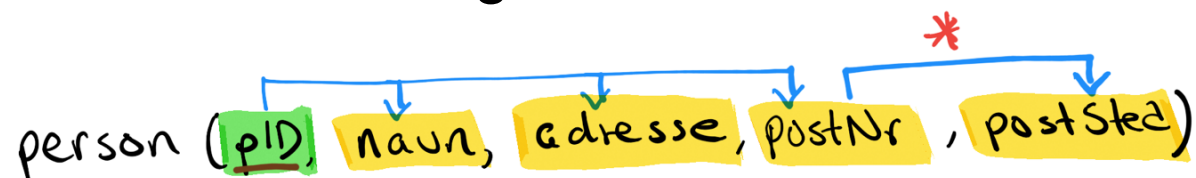
- Problem
 - postNr → postSted er en kilde til redundans
- Løsning
 - Splitte i to tabeller



- Begge tabeller er på 3NF
- Har fjernet kilde til redundans (f.a. mellom ikke-nøkkel-attributter)
- Men inter-tabell f.a.

Å oppnå tredje normalform (2)

- Alternativ løsning

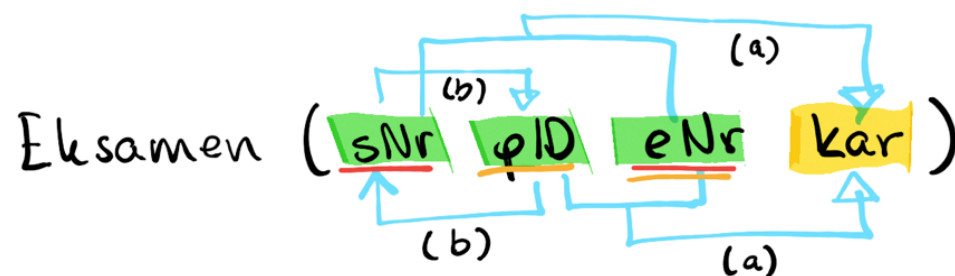


- Begge tabeller er på 3NF
- Unngår inter-tabell f.a.
- Bedre løsning

Boyce-Codd normalform

- 3NF kan ha redundansproblemer ved overlappende kandidatnøkler
- En tabell er på **Boyce-Codd normalform** (BCNF) hvis og bare hvis det for alle funksjonelle avhengigheter på formen, $X \rightarrow Y$, som gjelder for tabellen er slik at:
 - a) X er en supernøkkel i tabellen
- Alle venstresider i f.a.-er må altså være supernøkler (entydige identifikatorer for rader i tabellen)

Å oppnå BCNF



- Problem
 - $pID \rightarrow sNr$ og $sNr \rightarrow pID$ er kilder til redundans, selv om tabellen er på 3NF
- Løsning
 - Splitte i to tabeller



- Begge er på BCNF
- Har fjernet kilde til redundans
(avhengighet inne blant kandidatnøkklene)

Oppgave

- c) (4 %) Gitt $R = \{A, B, C, D, E\}$ og $F = \{A \rightarrow B, B \rightarrow C, CD \rightarrow E\}$. Gå ut fra at R oppfyller 1. normalform. Bestem den høyeste normalformen som oppfylles av R . Svaret må begrunnes.