

Normalisering

- ☐ Motivasjon
- ☐ Funksjonelle avhengigheter
- ☐ Determinanter
- ☐ Partielle avhengigheter
- ☐ Transitive avhengigheter
- ☐ Normalformer: 1NF, 2NF, 3NF, BCNF
- ☐ Normaliseringsstegene
- ☐ Denormalisering

Pensum: Kapittel 8.2

Problemer med dårlig design

KassaLappNr	VareNr	Antall	Betegnelse	Pris	Kategori
	13		Brunost	kr 107.00	Ost
	6		Crispisjok	kr 7.92	Sjokolade
6	8	0.97	Fårelabb	kr 235.00	Kjøttpålegg
	14		Gammalost	kr 120.00	Ost
	12		Gulost	kr 105.00	Ost
10	10	10	Leverpostei	kr 14.50	Kjøttpålegg
6	10	3	Leverpostei	kr 14.50	Kjøttpålegg
3	10	2	Leverpostei	kr 14.50	Kjøttpålegg
1	10	3	Leverpostei	kr 14.50	Kjøttpålegg

- ☐ Informasjon som ikke lar seg representere
- ☐ Redundans (oppdateringsanomalier)
- ☐ Nullmerker

Funksjonelle avhengigheter I

□ Eksempler:

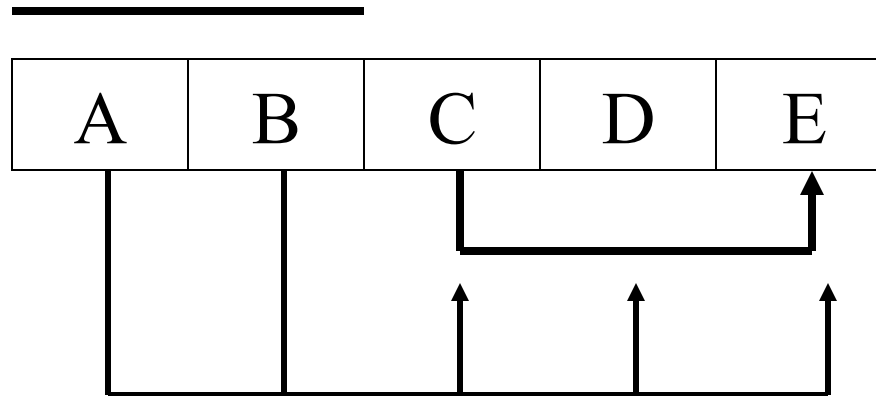
- $\text{VareNr} \rightarrow \text{Betegnelse}$
- $\text{VareNr} \rightarrow \text{Pris}$
- $\text{KassaLappNr}, \text{VareNr} \rightarrow \text{Antall}$

Funksjonelle avhengigheter II

- ❑ En funksjonell avhengighet $X \rightarrow Y$ uttrykker en beskrankning: Hvis to rader har samme verdi i X må de ha samme verdi i Y.
- ❑ Vi sier at X bestemmer Y, og kaller X en determinant.
- ❑ For å avgjøre om $X \rightarrow Y$ må vi kjenne *betydningen* til X og Y. Beskrankningen skal gjelde for *alle mulige* tabellinnhold.

Funksjonelle avhengigheter III

❑ Grafisk notasjon:



❑ X kandidatnøkkel: $X \rightarrow Y$ for alle Y

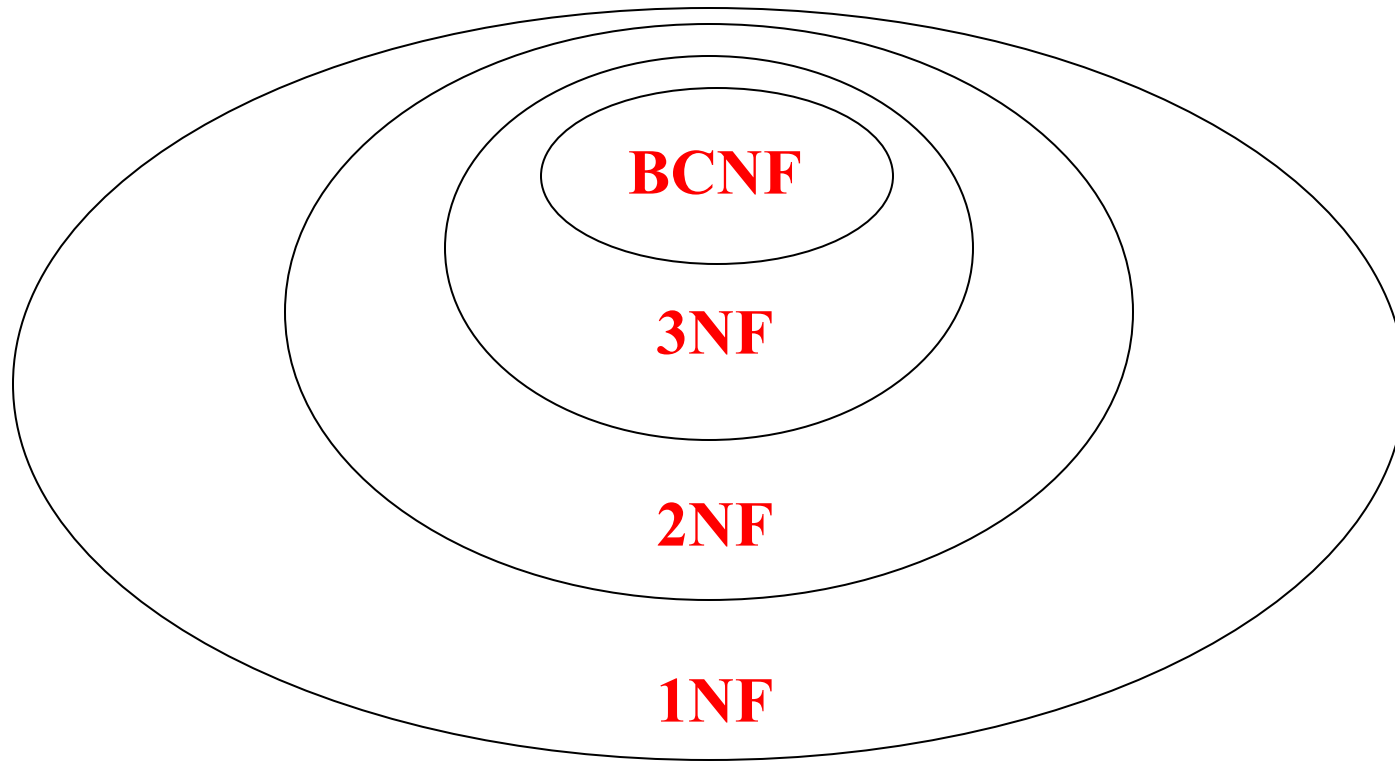
❑ Noen avhengigheter lar vi være underforstått.

Slutningsregler

- ❑ Hvis $X \rightarrow Y$ og $Y \rightarrow Z$, så $X \rightarrow Z$
- ❑ Hvis $X \rightarrow Y$, så $X, Z \rightarrow Y, Z$
- ❑ Noen funksjonelle avhengigheter kan utledes fra andre (ved hjelp av slutningsreglene).
- ❑ Eksempel:
 - Hvis $\text{AnsattNr} \rightarrow \text{PostNr}$ og $\text{PostNr} \rightarrow \text{PostSted}$, så $\text{AnsattNr} \rightarrow \text{PostSted}$.

Normalformer

Enhver tabell på BCNF er på 3NF, osv.

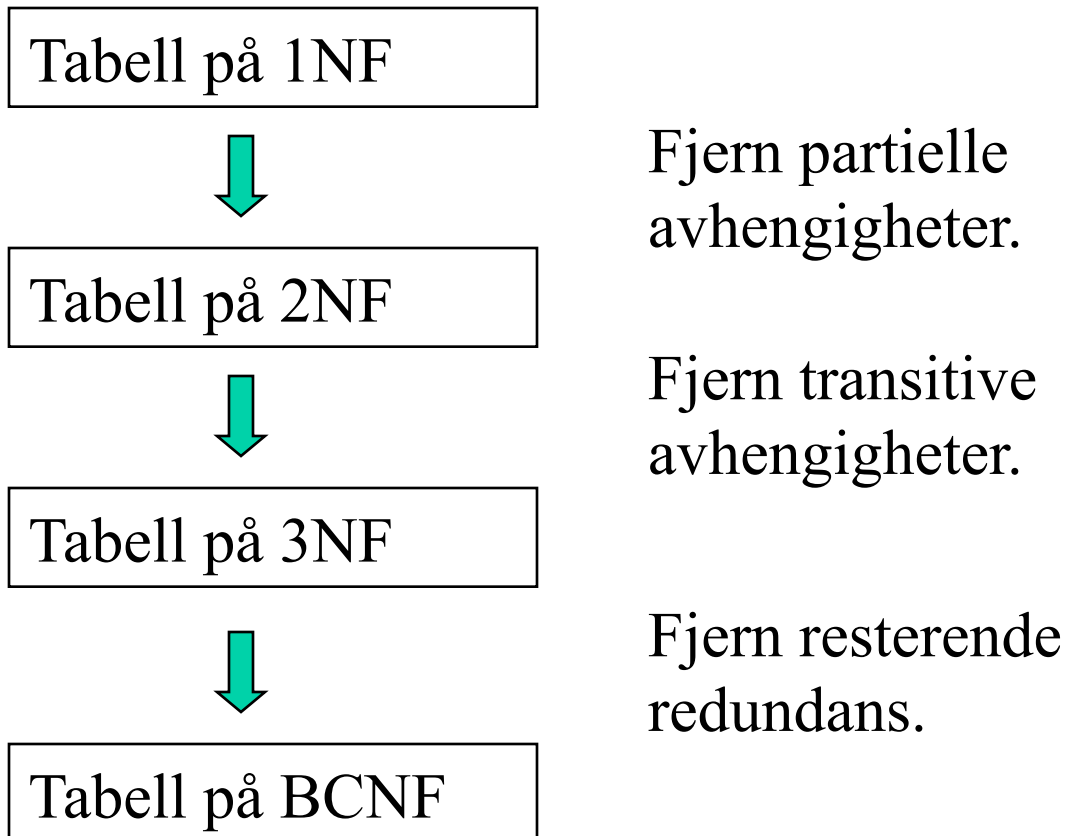


Normalisering

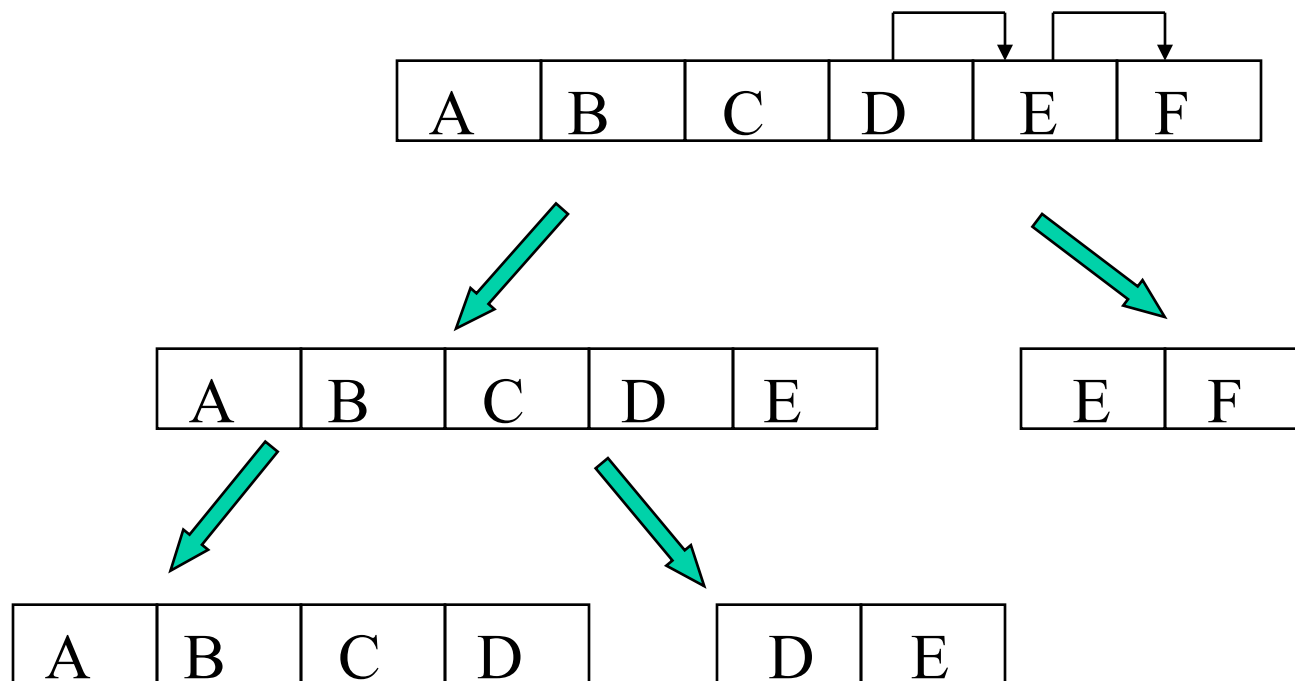
- ❑ Normalisering består i å dekomponere (splitte opp) tabeller til flere enklere tabeller slik at:
 - Redundans ikke oppstår
 - Informasjonsinnholdet bevares

- ❑ Normaliseringsteori gir oss:
 - Presise regler for å avgjøre om og hvordan tabeller skal dekomponeres basert på kunnskap om funksjonelle avhengigheter.

Normaliseringsprosessen



Normaliseringssteg



❑ For funksjonell avhengighet $E \rightarrow F$ i tabell T:

➤ E og F legges inn i ny tabell. Fjern F fra T.

1. NormalForm (1NF)

- ❑ En tabell tilfredsstillter 1NF hvis alle verdier er atomære.
 - 1NF er kun av historisk interesse; enhver tabell er pr. definisjon på 1NF.
 - Restriksjonen til atomære verdier hindrer ”tabeller i tabeller” / ”repeterende grupper”.
 - Vi har ikke jobbet med tabeller som ikke er på 1NF.

Brudd på første normalform

❑ Eksempel

Studentnr	Etternavn	Fornavn	Fag
225	Hansen	Ole	INF1000,INF1500 ,INF2000
335	Nilsen	Grete	INF1000,INF1300

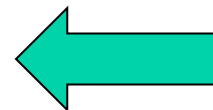
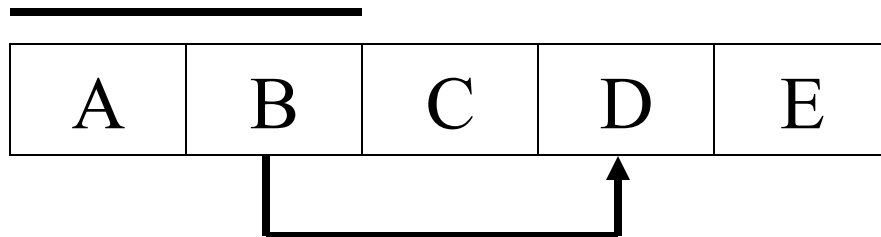
Løsning

Studentnr	Etternavn	Fornavn
225	Hansen	Ole
330	Nilsen	Grete

Studentnr	Fag
225	INF1000
225	INF1500
225	INF2000
330	INF1000
330	INF1300

2. NormalForm (2NF)

- ❑ En tabell tilfredsstiller 2NF hvis den er på 1NF og dessuten ikke har attributter som er partielt avhengig av primærnøkkelen.



Brudd!

Brudd på annen normalform

Studentnr	Fag	Fagnavn	Semester
225	INF1000	Databaser	H2007
330	INF1000	Databaser	V2008
225	INF1500	Operativsystemer	H2007
225	INF2000	Systemutvikling	V2008
330	INF1300	Dtmaskinarkitekt ur	H2006

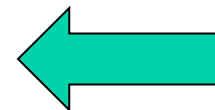
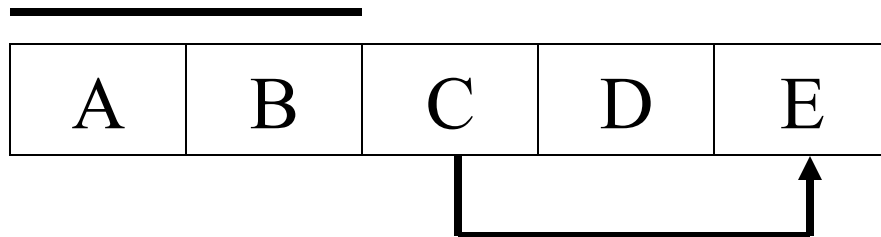
Løsning

Studentnr	Fag	Semester
225	INF1000	H2007
330	INF1000	V2008
225	INF1300	H2007
225	INF1500	V2008
330	INF2000	H2006

Fag	Fagnavn
INF1000	Databaser
INF1500	Operativsystemer
INF2000	Systemutvikling
INF1300	Datamaskinarkitektur

3. NormalForm (3NF)

- ❑ En tabell tilfredsstiller 3NF hvis den er på 2NF og dessuten ikke inneholder transitive funksjonelle avhengigheter.



Brudd!

Brudd på 3 NF

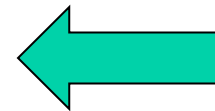
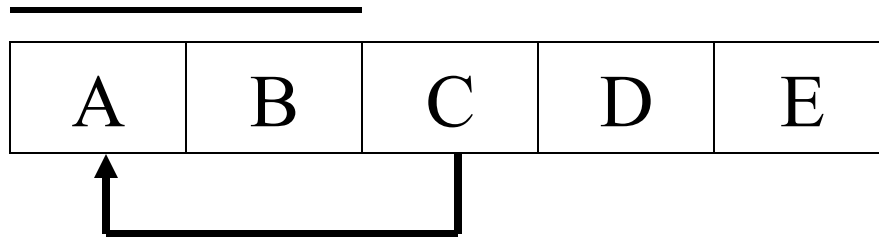
- ❑ Låner (LånerID, Navn, Adresse, Postnummer, Poststed)
- ❑ Har funksjonell avhengighet Postnr
→ Poststed

LånerID	Navn	Adresse	Postnr
1000	Hansen	Storgt	1003
1300	Olsen	Heia	7000

Postnr	Poststed
1003	OSLO
7000	TRONDHEIM

Boyce-Codd NormalForm (BCNF)

- ❑ En tabell er på BCNF hvis enhver minimal determinant er en kandidatnøkkel.



Brudd!

Brudd på BCNF

<u>Kinosal</u>	<u>Dato</u>	<u>Tidspunkt</u>	Filmnr
Sal 3	01/01/08	17:00	24
Sal 3	01/01/08	19:00	24

+ regel at en film alltid vises i samme kinosal

Løsning

<u>Dato</u>	<u>Tidspunkt</u>	<u>Filmnr</u>
01/01/08	17:00	24
01/01/08	19:00	24

<u>Filmnr</u>	Kinosal
24	Sal 3
17	Sal 2

Normaliseringseksempel I

StudNr	Snavn	KursKode	KursNavn	StPoeng	EksDato	Kar

1. Kartlegg funksjonelle avhengigheter.
2. Bestem kandidatnøkler / primærnøkkel.
3. Avgjør normalform.
4. Utfør normalisering til BCNF.
 - Gjenta steg 2-4 for hvert mellomresultat.

❑ Forslag til løsning:

Studnr → Snavn

Kurskode → Kursnavn, stp

Eksdato, Kurskode, Studnr → karakter

(Studnr, Snavn) (Kurskode, Kursnavn, stp)

(Eksdato, Kurskode, Studnr, karakter)

Normaliseringseksempel II

□ Gitt en tabell $T(A, B, C, D, E)$. Det er funksjonelle avhengigheter fra A til B, fra C til D og fra C til E.

- Bestem kandidatnøkkel.
- Hvilken normalform er tabellen på?
- Utfør normalisering til BCNF.

□ Det kan virke underlig at vi kan normalisere en tabell som inneholder data vi ikke vet hva er.

- Hvordan er dette mulig ?

Eksempel på normalisering

$T(A,B,C,D,E)$ $A \rightarrow B$, $C \rightarrow D$ og $C \rightarrow E$.

A og C er kandidatnøkkel siden de sammen bestemmer alle andre kolonner. Vi har flere partielle avhengigheter. Dette er et brudd på 2NF, så tabellen er på 1NF. Vi splitter T i to tabeller:

$T1(A,B,C)$ og $T2(C,D,E)$

$T2$ har C som kandidatnøkkel og er på BCNF.

I $T1$ så er $A+C$ kandidatnøkkel. $A \rightarrow B$ er en partiell avhengighet, så $T1$ er på 1NF. Vi splitter $T1$:

$T11(A,C)$ og $T12(A,B)$

Både $T11$ og $T12$ er på BCNF (om ikke annet så fordi de har kun to kolonner).

I $T12$ er A kandidatnøkkel. Sluttresultatet består av $T2$, $T11$ og $T12$.

Følgende tabell er et eksempel på hva den opprinnelige tabellen kan inneholde:

T(ProsjektNr, Budsjett, AnsattNr, Fornavn, Etternavn)

Resultatet blir, hvis vi omnavner T11, T12 og T2 til henholdsvis ProsjektDeltakelse, Prosjekt og Ansatt:

ProsjektDeltakelse(ProsjektNr*,AnsattNr*)

Prosjekt(ProsjektNr,Budsjett)

Ansatt(AnsattNr,Fornavn,Etternavn)

Avhengighets diagram (1NF)

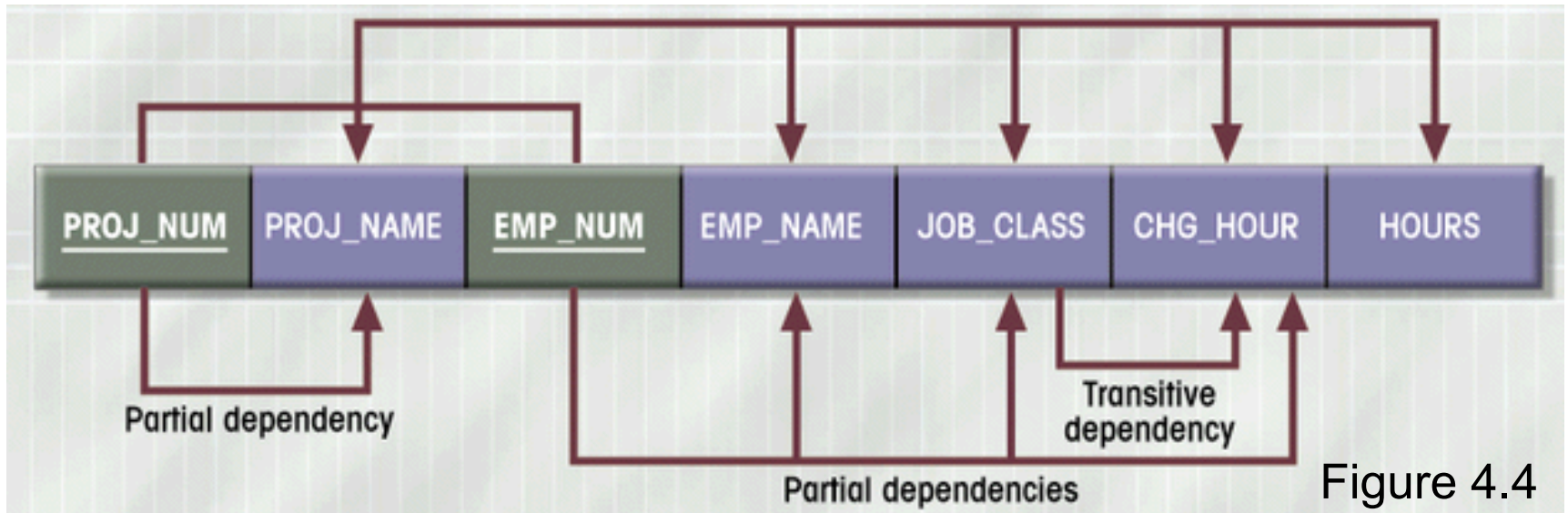
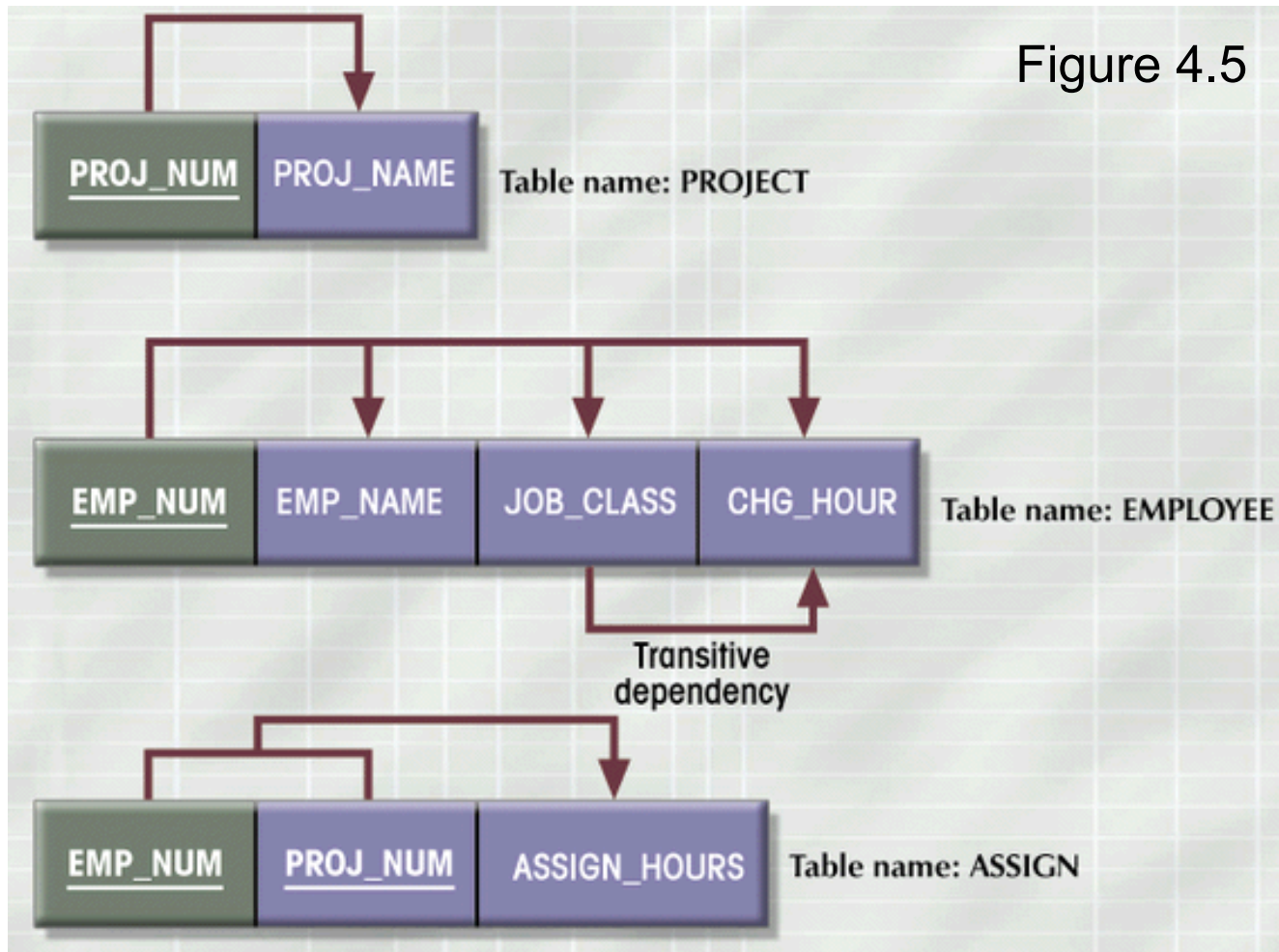


Figure 4.4

2NF



Eksempel 3NF

Figure 4.9a Elimination of
Transitive Dependency —
Relation with Transitive Dependency

HOUSING (SID, Dorm, Fee)

Key: SID

Functional

dependencies: Dorm \rightarrow Fee

SID \rightarrow Dorm \rightarrow Fee

SID	Dorm	Fee
100	Randolph	3200
150	Ingersoll	3100
200	Randolph	3200
250	Pitkin	3100
300	Randolph	3200

(a)

Her er tabellene på 3NF

Figure 4.9b Elimination of Transitive Dependency —
Relations Eliminating the Transitive Dependency

STU-HOUSING (SID, Dorm)

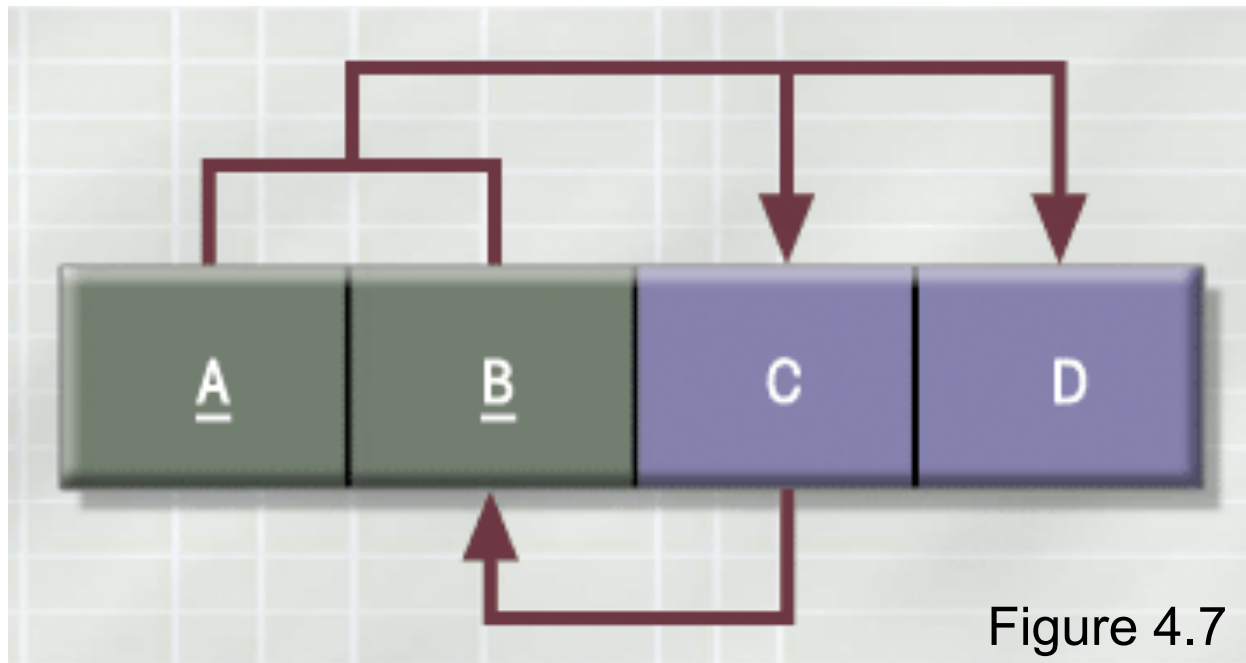
SID	Dorm
100	Randolph
150	Ingersoll
200	Randolph
250	Pitkin
300	Randolph

BLDG-FEE (Dorm, Fee)

Dorm	Fee
Randolph	3200
Ingersoll	3100
Pitkin	3100

(b)

3NF Table Not in BCNF



Dekomposisjon for å oppnå BCNF

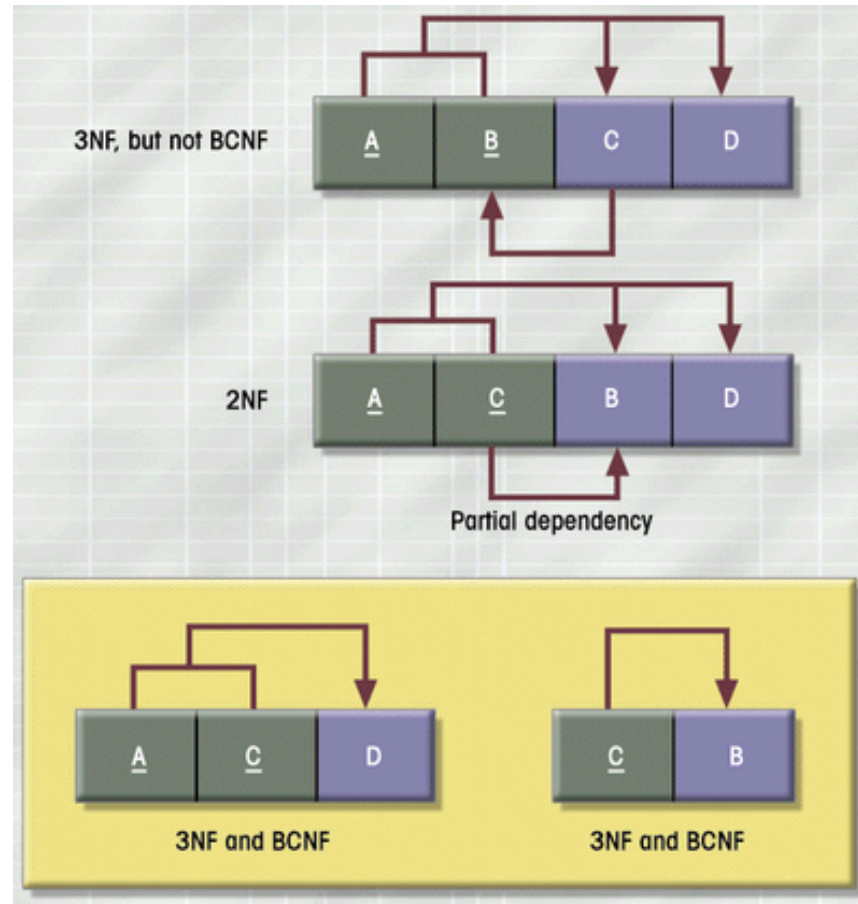


Figure 4.8

Ikke på 4 NF

Figure 4.11 Relation with Multi-Value Dependencies

STUDENT (SID, Major, Activity)

Multi-value

dependencies: $SID \twoheadrightarrow Major$
 $SID \twoheadrightarrow Activity$

SID	Major	Activity
100	Music	Swimming
100	Accounting	Swimming
100	Music	Tennis
100	Accounting	Tennis
150	Math	Jogging

På 4 NF

Figure 4.13 Elimination of Multi-Value Dependency

STU-MAJOR (SID, Major)

SID	Major
100	Music
100	Accounting
150	Math

STU-ACT (SID, Activity)

SID	Activity
100	Skiing
100	Swimming
100	Tennis
150	Jogging

E/R og normalisering

- ❑ Er normalisering et alternativ til E/R? Mulig strategi:
 1. Start med "universalrelasjonen" (alt i èn tabell).
 2. Kartlegg funksjonelle avhengigheter.
 3. Utfør normalisering.
- ❑ Entiteter med attributter er en naturlig tenkemåte?
 - E/R kan brukes for å kommunisere med brukere.
- ❑ Normalisering kan brukes for å sjekke en datamodell.
 - Dette er en jobb for databasedesigneren.

Denormalisering

- ❑ Normalisering gir flere tabeller.
 - Flere tabeller gir flere likekoblinger.
 - Sikring av visse forretningsregler krever at DBHS ser i flere tabeller.
- ❑ Av effektivitetshensyn kan vi i visse tilfeller forsvare å slå sammen normaliserte tabeller.
- ❑ Dette kalles for denormalisering, og gir en form for ”kontrollert redundans”.
- ❑ Normalisering fra 3NF til BCNF gir forretningsregler som er tunge for DBHS å sikre. Det er dermed ikke uvanlig å stoppe normalisering ved 3NF.