

IN2090 – Databaser og datamodellering

08 – Bestemme normalform

Leif Harald Karlsen
leifhka@ifi.uio.no



Universitetet i Oslo

Hvordan sjekke normalformer

- ◆ For å sjekke hvilken NF et skjema oppfyller, kan vi sjekke alle tabeller opp mot alle FDer
- ◆ Og deretter sjekke om FDen bryter med BCNF, 3NF og 2NF
- ◆ Vi skal nå se en algoritme som gjør dette systematisk
- ◆ Det er derimot to ting vi må gjøre med FDene før vi kan bruke algoritmen: Skrive om FDer og fjerne redundans

Skrive om FDer

- ◆ FDene må skrives på formen $X \rightarrow A$ (splitt høyresidene).
- ◆ F.eks. fremfor

ProduktID \rightarrow Navn, Pris

må man skrive

ProduktID \rightarrow Navn

ProduktID \rightarrow Pris

- ◆ Algoritmen antar altså at det kun er én attributt på høyresidene

Fjerne redundans

- ◆ Hvis $X \rightarrow B$, så har vi også $X \cup Y \rightarrow B$ for alle Y
- ◆ F.eks. dersom

ProduktID \rightarrow Navn

så har vi jo også

ProduktID, Pris \rightarrow Navn

men ønsker kun å bruke den øverste

- ◆ Vi må fjerne alle redundante FDer, algoritmen antar ingen redundans
- ◆ Må gjøres når man bestemmer FDene

Algoritme for å sjekke NF

- ◆ Først, finn alle kandidatnøkler med algoritmen fra forrige uke
- ◆ For hver tabell og hver FD $X \rightarrow A$:
 1. Er X en supernøkkel?
 - Ja: BCNF så langt, gå til neste FD
 - Nei: brudd på BCNF. Gå til 2.
 2. Er A et nøkkelattributt?
 - Ja: 3NF så langt, gå til neste FD
 - Nei: brudd på 3NF. Gå til 3.
 3. Er X del av en kandidatnøkkel?
 - Nei: 2NF så langt, gå til neste FD
 - Ja: brudd på 2NF og skjema er på 1NF, stopp.
- ◆ Tabellen er så på den laveste normalformen vi får ut av denne algoritmen
- ◆ Skjemaet er på den laveste normalformen av tabellenes
- ◆ Med andre ord: Hvis jeg har en tabell og en FD som bryter 2NF, er skjemaet på 1NF.

Eksempel 1

Algoritme:

Finn alle kandidatnøkler.

For hver tabell og hver FD $X \rightarrow A$:

1. Er X en supernøkkel?

Ja: BCNF så langt, gå til neste FD

Nei: brudd på BCNF. Gå til 2.

2. Er A et nøkkelattributt?

Ja: 3NF så langt, gå til neste FD

Nei: brudd på 3NF. Gå til 3.

3. Er X del av en kandidatnøkkel?

Nei: 2NF så langt, gå til neste FD

Ja: brudd på 2NF og skjema er på 1NF, stopp.

Finn normalformen:

$S(\text{Brnavn}, \text{Navn}, \text{Etternavn}, \text{Kurskode}, \text{KursTittel}, \text{Karakter})$

FDer:

1. $\text{Brnavn}, \text{Kurskode} \rightarrow \text{Karakter}$

2. $\text{Brnavn} \rightarrow \text{Navn}$

3. $\text{Brnavn} \rightarrow \text{Etternavn}$

4. $\text{Kurskode} \rightarrow \text{KursTittel}$

Kandidatnøkkel: $\{\text{Brnavn}, \text{Kurskode}\}$.

- ◆ FD 1: $\text{Brnavn}, \text{Kurskode}$ er en supernøkkel, så BCNF så langt
- ◆ FD 2: Brnavn ikke supernøkkel, brudd på BCNF
- ◆ FD 2: Navn ikke nøkkelattributt, brudd på 3NF
- ◆ FD 2: Brnavn del av en kandidatnøkkel, brudd på 2NF

Relasjonen S er derfor på 1NF.

Eksempel 2

Algoritme:

Finn alle kandidatnøkler.

For hver tabell og hver FD $X \rightarrow A$:

1. Er X en supernøkkel?

Ja: BCNF så langt, gå til neste FD

Nei: brudd på BCNF. Gå til 2.

2. Er A et nøkkelattributt?

Ja: 3NF så langt, gå til neste FD

Nei: brudd på 3NF. Gå til 3.

3. Er X del av en kandidatnøkkel?

Nei: 2NF så langt, gå til neste FD

Ja: brudd på 2NF og skjema er på 1NF, stopp.

Finn normalformen:

Produkt($pid, navn, kategori, pris, butikk$)

FDer:

1. $navn, kategori \rightarrow pid$

2. $pid, butikk \rightarrow pris$

3. $pid \rightarrow kategori$

4. $pid \rightarrow navn$

Kandidatnøkler: $\{pid, butikk\}, \{navn, kategori, butikk\}$.

FD 1:

1. $\{navn, kategori\}$ er ikke en supernøkkel, så brudd på BCNF.

2. pid er nøkkelattributt, så 3NF så langt.

Eksempel 2

Algoritme:

Finn alle kandidatnøkler.

For hver tabell og hver FD $X \rightarrow A$:

1. Er X en supernøkkel?

Ja: BCNF så langt, gå til neste FD

Nei: brudd på BCNF. Gå til 2.

2. Er A et nøkkelattributt?

Ja: 3NF så langt, gå til neste FD

Nei: brudd på 3NF. Gå til 3.

3. Er X del av en kandidatnøkkel?

Nei: 2NF så langt, gå til neste FD

Ja: brudd på 2NF og skjema er på 1NF, stopp.

Finn normalformen:

Produkt(pid, navn, kategori, pris, butikk)

FDer:

1. navn, kategori \rightarrow pid

2. pid, butikk \rightarrow pris

3. pid \rightarrow kategori

4. pid \rightarrow navn

Kandidatnøkler: {pid, butikk}, {navn, kategori, butikk}.

FD 2:

1. {pid, butikk} er en supernøkkel, så BCNF så langt.

Eksempel 2

Algoritme:

Finn alle kandidatnøkler.

For hver tabell og hver FD $X \rightarrow A$:

1. Er X en supernøkkel?

Ja: BCNF så langt, gå til neste FD

Nei: brudd på BCNF. Gå til 2.

2. Er A et nøkkelattributt?

Ja: 3NF så langt, gå til neste FD

Nei: brudd på 3NF. Gå til 3.

3. Er X del av en kandidatnøkkel?

Nei: 2NF så langt, gå til neste FD

Ja: brudd på 2NF og skjema er på 1NF, stopp.

Finn normalformen:

Produkt(pid, navn, kategori, pris, butikk)

FDer:

1. navn, kategori \rightarrow pid

2. pid, butikk \rightarrow pris

3. pid \rightarrow kategori

4. pid \rightarrow navn

Kandidatnøkler: {pid, butikk}, {navn, kategori, butikk}.

FD 3:

1. pid er ikke en supernøkkel, så brudd på BCNF.

2. kategori er en nøkkelattributt, så 3NF så langt.

Eksempel 2

Algoritme:

Finn alle kandidatnøkler.

For hver tabell og hver FD $X \rightarrow A$:

1. Er X en supernøkkel?

Ja: BCNF så langt, gå til neste FD

Nei: brudd på BCNF. Gå til 2.

2. Er A et nøkkelattributt?

Ja: 3NF så langt, gå til neste FD

Nei: brudd på 3NF. Gå til 3.

3. Er X del av en kandidatnøkkel?

Nei: 2NF så langt, gå til neste FD

Ja: brudd på 2NF og skjema er på 1NF, stopp.

Finn normalformen:

Produkt(pid, navn, kategori, pris, butikk)

FDer:

1. navn, kategori \rightarrow pid

2. pid, butikk \rightarrow pris

3. pid \rightarrow kategori

4. pid \rightarrow navn

Kandidatnøkler: {pid, butikk}, {navn, kategori, butikk}.

FD 4:

1. pid er ikke en supernøkkel, så brudd på BCNF.

2. navn er et nøkkelattributt, så ikke brudd på 3NF

Altså er Produkt på 3NF.

Takk for nå!

Neste video vil handle om tapsfri dekomposisjon.