

IN2090 – Databaser og datamodellering

08 – Eksempler: Databasesdesign og normalformer

Leif Harald Karlsen (Evgenij Thorstensen)

leifhka@ifi.uio.no



Universitetet i Oslo

Oppgave 1 – Løsning

Gitt følgende relasjon:

Person(personnr, navn, initialer, fødselsdato, alder)

med FDene:

- ◆ personnr \rightarrow navn, fødselsdato
- ◆ navn \rightarrow initialer
- ◆ fødselsdato \rightarrow alder

Kandidatnøkler (fra forrige uke): personnr

1. Finn ut hvilken normalform relasjonen er på
2. Dekomponer relasjonen til BCNF

Normalform:

- ◆ personnr \rightarrow navn, fødselsdato: bryter ikke med BCNF
- ◆ navn \rightarrow initialer:
 - ◆ Bryter med BCNF (navn ikke supernøkkel);
 - ◆ bryter med 3NF (initialer ikke nøkkelattributt);
 - ◆ bryter ikke med 2NF (navn ikke del av kandidatnøkkel).
- ◆ fødselsdato \rightarrow alder:
 - ◆ Bryter med BCNF (fødselsdato ikke supernøkkel);
 - ◆ bryter med 3NF (alder ikke nøkkelattributt);
 - ◆ bryter ikke med 2NF (fødselsdato ikke del av kandidatnøkkel).

Altså er Person på 2NF.

Oppgave 1 – Løsning

Relasjon: Person(personnr, navn, initialer, født, alder)

med FDene:

1. personnr \rightarrow navn, født
2. navn \rightarrow initialer
3. født \rightarrow alder

Gjør følgende:

1. Finn kandidatnøkler: personnr
2. Finn normalform: 2NF
3. Dekomponer til BCNF

Dekomponering:

- ◆ navn \rightarrow initialer: Bryter BCNF
 - ◆ navn⁺ = {navn, initialer}
 - ◆ Dekomponerer til:
S₁(navn, initialer) (FD: 2 – KN: {navn})
S₂(personnr, navn, født, alder) (FD: 1, 3 – KN: {personnr})
- ◆ S₁: FDen bryter ikke BCNF
- ◆ S₂: født \rightarrow alder bryter BCNF
 - ◆ født⁺ = {født, alder}
 - ◆ Dekomponerer til:
S₂₁(født, alder) (FD: 3 – KN: {født})
S₂₂(personnr, navn, født) (FD: 1 – KN: {personnr})
- ◆ Hverken S₂₁ eller S₂₂ bryter BCNF

Person dekomponeres altså tapsfritt til BCNF med:

- ◆ S₁(navn, initialer) (FD: 2 – KN: {navn})
- ◆ S₂₁(født, alder) (FD: 3 – KN: {født})
- ◆ S₂₂(personnr, navn, født) (FD: 1 – KN: {personnr})

Oppgave 2 – Løsning

Produkt(produktID, navn, kategori, pris, butikkID, butikknavn, butikktype, adresse, postnr, poststed)

med FDene:

- ◆ produktID \rightarrow navn
 - ◆ produktID \rightarrow kategori
 - ◆ produktID \rightarrow pris
 - ◆ navn, kategori \rightarrow produktID
 - ◆ butikkID \rightarrow butikknavn
 - ◆ butikkID \rightarrow butikktype
 - ◆ butikkID \rightarrow adresse
 - ◆ butikkID \rightarrow postnr
 - ◆ postnr \rightarrow poststed
1. Kandidatnøkler: {butikkID, produktID}, {butikkID, navn, kategori}
 2. Finn normalformen: 1NF
 3. Dekomponer til BCNF

Normalform:

- ◆ produktID \rightarrow navn:
 - ◆ Bryter BCNF (produktID ikke supernøkkel)
 - ◆ Bryter ikke 3NF (navn er nøkkelattributt)
- ◆ produktID \rightarrow kategori: Samme som over
- ◆ produktID \rightarrow pris:
 - ◆ Bryter BCNF (produktID ikke supernøkkel)
 - ◆ Bryter 3NF (pris ikke nøkkelattributt)
 - ◆ Bryter 2NF (produktID er del av kandidatnøkkel)

Altså er Produkt på 1NF.

Oppgave 2 – Løsning

Produkt(produktID, navn, kategori, pris, butikkID, butikknavn, butikktype, adresse, postnr, poststed)

med FDene:

1. produktID \rightarrow navn
2. produktID \rightarrow kategori
3. produktID \rightarrow pris
4. navn, kategori \rightarrow produktID
5. butikkID \rightarrow butikknavn
6. butikkID \rightarrow butikktype
7. butikkID \rightarrow adresse
8. butikkID \rightarrow postnr
9. postnr \rightarrow poststed

Utfør:

1. Finn kandiadt nøkler:
{butikkID, produktID},
{butikkID, navn, kategori}
2. Finn normalformen til relasjonen (forrige foil): 1NF
3. Dekomponer relasjonen til BCNF

Dekomponering av Produkt:

- ◆ produktID \rightarrow navn: Bryter BCNF
 - ◆ produktID⁺ = {produktID, navn, kategori, pris}
 - ◆ Dekomponerer til:
S₁(produktID, navn, kategori, pris)
S₂(produktID, butikkID, butikknavn,
butikktype, adresse, postnr, poststed)

Oppgave 2 – Løsning

$S_1(\text{produktID}, \text{navn}, \text{kategori}, \text{pris})$

med FDene:

1. $\text{produktID} \rightarrow \text{navn}$
2. $\text{produktID} \rightarrow \text{kategori}$
3. $\text{produktID} \rightarrow \text{pris}$
4. $\text{navn}, \text{kategori} \rightarrow \text{produktID}$
5. $\text{butikkID} \rightarrow \text{butikknavn}$
6. $\text{butikkID} \rightarrow \text{butikktype}$
7. $\text{butikkID} \rightarrow \text{adresse}$
8. $\text{butikkID} \rightarrow \text{postnr}$
9. $\text{postnr} \rightarrow \text{poststed}$

Utfør:

1. Finn kandidatnøkler: $\{\text{produktID}\}, \{\text{navn}, \text{kategori}\}$
2. Finn normalformen til relasjonen: BCNF
3. Trenger ikke dekomponere S_1 videre

Oppgave 2 – Løsning

$S_2(\text{produktID}, \text{butikkID}, \text{butikknavn}, \text{butikktype}, \text{adresse}, \text{postnr}, \text{poststed})$

med FDene:

1. $\text{produktID} \rightarrow \text{navn}$
2. $\text{produktID} \rightarrow \text{kategori}$
3. $\text{produktID} \rightarrow \text{pris}$
4. $\text{navn}, \text{kategori} \rightarrow \text{produktID}$
5. $\text{butikkID} \rightarrow \text{butikknavn}$
6. $\text{butikkID} \rightarrow \text{butikktype}$
7. $\text{butikkID} \rightarrow \text{adresse}$
8. $\text{butikkID} \rightarrow \text{postnr}$
9. $\text{postnr} \rightarrow \text{poststed}$

Dekomponering av S_2 :

- ◆ $\text{butikkID} \rightarrow \text{butikknavn}$: Bryter BCNF
 - ◆ $\text{butikkID}^+ = \{\text{butikkID}, \text{butikknavn}, \text{butikktype}, \text{adresse}, \text{postnr}, \text{poststed}\}$
 - ◆ Dekomponerer til:
 $S_{21}(\text{butikkID}, \text{butikknavn}, \text{butikktype}, \text{adresse}, \text{postnr}, \text{poststed})$
 $S_{22}(\text{produktID}, \text{butikkID})$ (FD: - KN: $\{\text{produktID}, \text{butikkID}\}$)(BCNF)

Utfør:

1. Finn kandidatnøkler:
 $\{\text{produktID}, \text{butikkID}\}$
2. Finn normalformen til relasjonen: 1NF
3. Dekomponer relasjonen til BCNF

Oppgave 2 – Løsning

$S_{21}(\text{butikkID}, \text{butikknavn}, \text{butikktype}, \text{adresse}, \text{postnr}, \text{poststed})$

med FDene:

1. $\text{produktID} \rightarrow \text{navn}$
2. $\text{produktID} \rightarrow \text{kategori}$
3. $\text{produktID} \rightarrow \text{pris}$
4. $\text{navn}, \text{kategori} \rightarrow \text{produktID}$
5. $\text{butikkID} \rightarrow \text{butikknavn}$
6. $\text{butikkID} \rightarrow \text{butikktype}$
7. $\text{butikkID} \rightarrow \text{adresse}$
8. $\text{butikkID} \rightarrow \text{postnr}$
9. $\text{postnr} \rightarrow \text{poststed}$

Dekomponering av S_{21} :

- ◆ $\text{postnr} \rightarrow \text{poststed}$: Bryter BCNF
 - ◆ $\text{postnr}^+ = \{\text{postnr}, \text{poststed}\}$
 - ◆ Dekomponerer til disse:
 - $S_{211}(\text{postnr}, \text{poststed})$ (FD: 9, KN: $\{\text{postnr}\}$)
 - $S_{212}(\text{butikkID}, \text{butikknavn}, \text{butikktype}, \text{adresse}, \text{postnr})$ (FD: 5-8, KN: $\{\text{butikkID}\}$)
- ◆ Begge disse er på BCNF

Utfør:

1. Finn kandidatnøkler:
 $\{\text{butikkID}\}$
2. Finn normalformen til relasjonen: 2NF
3. Dekomponer relasjonen til BCNF

Oppgave 2 – Løsning

Produkt(produktID, navn, kategori, pris, butikkID, butikknavn, butikktype, adresse, postnr, poststed)

med FDene:

1. produktID \rightarrow navn
2. produktID \rightarrow kategori
3. produktID \rightarrow pris
4. navn, kategori \rightarrow produktID
5. butikkID \rightarrow butikknavn
6. butikkID \rightarrow butikktype
7. butikkID \rightarrow adresse
8. butikkID \rightarrow postnr
9. postnr \rightarrow poststed

Dekomponering blir altså:

$S_1(\text{produktID}, \text{navn}, \text{kategori}, \text{pris})$

$S_{211}(\text{postnr}, \text{poststed})$

$S_{212}(\text{butikkID}, \text{butikknavn}, \text{butikktype}, \text{adresse}, \text{postnr})$

$S_{22}(\text{produktID}, \text{butikkID})$

med FDer og kandidatnøkler som vist på de andre foilene.

Oppgave 3 (Vanskelig, ikke pensum!)

Lag et skjema på BCNF som inneholder dataene for 2022 i "Fisketillatelser med fartøytilknytning og kvotestørrelser" fra Fiskeridirektoratet:

<https://www.fiskeridir.no/Tall-og-analyse/AApne-data/Fiskere-fartoe-y-og-fisketillatelser>

Oppgave 3 – Løsning: Lage tabell

- ◆ Starter med å laste ned filen og åpne i et regnearkprogram (e.g. Libreoffice) og åpner arket med navn "2022"
- ◆ Lagrer filen som CSV med semicolon som "delimiter" og all tekst omringet av enkle (') fnutter (filnavn kvoter.csv)
- ◆ Flytter øverste linje fra CSV-filen inn i egen fil og skriver det om til en `CREATE TABLE`-kommando:

```
CREATE SCHEMA fiskeri;  
CREATE TABLE fiskeri.kvoter_raw (  
    Data_pr text,  
    Fartøy_ID text,  
    Registreringsmerke text,  
    Tillatelse_kode text,  
    Tillatelse text,  
    Tillatelse_ID text,  
    Tillatelse_fra_dato text,  
    Tillatelse_til_dato text,  
    Linjenummer int,  
    Linjenummer_beskrivelse text,  
    Kvotestørrelse float,  
    Kvotestr_fra_dato text,  
    Kvotestr_til_dato text  
);
```

Oppgave 3 – Løsning: Datainnlasting

- ◆ Datoene er på feil (norsk) format, må oversettes til noe PostgreSQL-forstår
- ◆ Gjør dette med et view, regexp_replace og casting til riktig type:

```
CREATE VIEW fiskeri.kvoter AS
SELECT
  regexp_replace(Data_pr, '(\d\d)\.(\d\d)\.(\d\d\d\d)(.*)', '\3-\2-\1\4')::date AS Data_pr,
  Fartøy_ID,
  Registreringsmerke,
  Tillatelse_kode,
  Tillatelse,
  Tillatelse_ID,
  regexp_replace(Tillatelse_fra_dato, '(\d\d)\.(\d\d)\.(\d\d\d\d)(.*)', '\3-\2-\1\4')::timestamp
    AS Tillatelse_fra_dato,
  regexp_replace(Tillatelse_til_dato, '(\d\d)\.(\d\d)\.(\d\d\d\d)(.*)', '\3-\2-\1\4')::timestamp
    AS Tillatelse_til_dato,
  Linjenummer,
  Linjenummer_beskrivelse,
  Kvotestørrelse,
  regexp_replace(Kvotestr_fra_dato, '(\d\d)\.(\d\d)\.(\d\d\d\d)(.*)', '\3-\2-\1\4')::date
    AS Kvotestr_fra_dato,
  regexp_replace(Kvotestr_til_dato, '(\d\d)\.(\d\d)\.(\d\d\d\d)(.*)', '\3-\2-\1\4')::date
    AS Kvotestr_til_dato
FROM fiskeri.kvoter_raw;
```

Oppgave 3 – Løsning: Datainnlasting

Kjører følgende kommando for å laste dataene inn i tabellen vår:

```
cat kvoter.csv | psql <flagg> -c "COPY fiskeri.kvoter_raw FROM stdin CSV DELIMITER ';' NULL AS ''";
```

hvor <flagg> er de vanlige tilkoblingsdetaljene til den personlige databasen

Oppgave 3 – Løsning: Bestemme FDer og dekomponering

FDer (fra dokumentasjonen om dataene):

1. Fartøy_ID → Registreringsmerke
2. Tillatelse_kode → Tillatelse
3. Tillatelse_ID → Data_pr
4. Tillatelse_ID → Fartøy_ID
5. Tillatelse_ID → Tillatelse_kode
6. Tillatelse_ID → Tillatelse_Gjelder_Fra_Dato
7. Tillatelse_ID → Tillatelse_Gjelder_Til_Dato
8. Tillatelse_ID → Kvotestr_Gjelder_Fra_Dato
9. Tillatelse_ID → Kvotestr_Gjelder_Til_Dato
10. Tillatelse_ID, Linjenummer → Linjenummer_beskrivelse
11. Tillatelse_ID, Linjenummer → Kvotestørrelse

Kandidatnøkkel: {TillatelseID, Linjenummer}

Oppgave 3 – Løsning: Dekomponering

```
BEGIN;
```

```
-- Fartøy_ID -> Registreringsmerke bryter med BCNF  
-- Tillukningen til Fartøy_ID er {Fartøy_ID, Registreringsmerke},  
-- altså får vi følgende (som ikke bryter med BCNF):
```

```
CREATE TABLE fiskeri.fartøy(  
    Fartøy_ID text PRIMARY KEY,  
    Registreringsmerke text  
);
```

Oppgave 3 – Løsning: Dekomponering

```
-- Har nå en tabell med alle attributter bortsett fra Registreringsmerke
-- Tillatelse_kode -> Tillatelse bryter med BCNF
-- Tillukningen til Tillatelse_kode er {Tillatelse_kode, Tillatelse},
-- så får følgende (som ikke bryter BCNF):
CREATE TABLE fiskeri.tillatelsesInfo(
    Tillatelse_kode text PRIMARY KEY,
    Tillatelse text
);
```


Oppgave 3 – Løsning: Dekomponering

```
-- Har nå en tabell med alle attributter bortsett fra Registreringsmerke og Tillatelse
-- Tillatelse_ID -> Data_pr bryter med BCNF
-- Tillukningen til Tillatelse_ID er
--   {Tillatelse_ID, Data_pr, Fartøy_ID, Tillatelse_kode,
--     Tillatelse_Gjelder_Fra_Dato, Tillatelse_Gjelder_Til_Dato,
--     Kvotestr_Gjelder_Fra_Dato, Kvotestr_Gjelder_Til_Dato},
-- så får følgende (som ikke bryter BCNF):
CREATE TABLE fiskeri.tillatelse(
  Tillatelse_ID text PRIMARY KEY,
  Data_pr date,
  Fartøy_ID text REFERENCES fiskeri.fartøy(Fartøy_ID),
  Tillatelse_kode text REFERENCES fiskeri.tillatelsesInfo(Tillatelse_kode),
  Tillatelse_fra_dato timestamp,
  Tillatelse_til_dato timestamp,
  Kvotestr_fra_dato timestamp,
  Kvotestr_til_dato timestamp
);
```

Oppgave 3 – Løsning: Dekomponering

```
-- Står nå igjen med følgende tabell, som ikke bryter med BCNF:  
CREATE TABLE fiskeri.tillatelsesLinje(  
    Tillatelse_ID text REFERENCES fiskeri.tillatelse(Tillatelse_ID),  
    Linjenummer int,  
    Linjenummer_beskrivelse text,  
    Kvotestørrelse float,  
    PRIMARY KEY (Tillatelse_ID, Linjenummer)  
);
```

Oppgave 3 – Løsning: Migrering

```
-- Setter så inn data fra fiskeri.kvoter i hver tabell:
```

```
INSERT INTO fiskeri.fartøy
SELECT DISTINCT Fartøy_ID, Registreringsmerke
FROM fiskeri.kvoter;
```

```
INSERT INTO fiskeri.tillatelsesInfo
SELECT DISTINCT Tillatelse_kode, Tillatelse
FROM fiskeri.kvoter;
```

```
INSERT INTO fiskeri.tillatelse
SELECT DISTINCT Tillatelse_ID, Data_pr, Fartøy_ID, Tillatelse_kode,
    Tillatelse_fra_dato, Tillatelse_til_dato
    Kvotestr_fra_dato, Kvotestr_til_dato
FROM fiskeri.kvoter;
```

```
INSERT INTO fiskeri.tillatelsesLinje
SELECT DISTINCT Tillatelse_ID, Linjenummer, Linjenummer_beskrivelse, Kvotestørrelse
FROM fiskeri.kvoter
WHERE Linjenummer IS NOT NULL; -- Finnes rader i kvoter som mangler linjenummer
```