

# 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

---

Gitt følgende relasjon:

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

med FDene:

- ◆  $\text{personnr} \rightarrow \text{navn, fødselsdato}$
- ◆  $\text{navn} \rightarrow \text{initialer}$
- ◆  $\text{fødselsdato} \rightarrow \text{alder}$

Kandidatnøkler (fra forrige uke): personnr

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

**Dekomponering:**

◆  $\text{navn} \rightarrow \text{initialer}$ : Bryter BCNF

- ◆  $\text{navn}^+ = \{\text{navn, initialer}\}$
- ◆ Dekomponerer til  $S_1(\text{navn, initialer})$  (på BCNF) og  $S_2(\text{personnr, navn, fødselsdato, alder})$

◆  $\text{fødselsdato} \rightarrow \text{alder}$ : Bryter BCNF for  $S_2$

- ◆  $\text{fødselsdato}^+ = \{\text{fødselsdato, alder}\}$
- ◆ Dekomponerer til  $S_{21}(\text{fødselsdato, alder})$  (på BCNF) og  $S_{22}(\text{personnr, navn, fødselsdato})$  (på BCNF)

Person dekomponeres altså tapsfritt til BCNF med  $S_1(\text{navn, initialer})$ ,  $S_{21}(\text{fødselsdato, alder})$  og  $S_{22}(\text{personnr, navn, fødselsdato})$ .

# 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

Kandidatnøkler (fra forrige uke): {butikkID, produktID},  
{butikkID, navn, kategori}

1. Finn normalformen til relasjonen
2. Dekomponer relasjonen 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:

- ◆ 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

Kandiadtnøkler (fra forige uke):

{butikkID, produktID},  
{butikkID, navn, kategori}

1. Finn normalformen til relasjonen
2. Dekomponer relasjonen til BCNF

**Dekomponering av Produkt:**

- ◆ produktID  $\rightarrow$  navn: Bryter BCNF
  - ◆ produktID<sup>+</sup> = {produktID, navn, kategori, pris}
  - ◆ Dekomponerer til S<sub>1</sub>(produktID, navn, kategori, pris) (på BCNF) og S<sub>2</sub>(produktID, butikkID, butikknavn, butikktype, adresse, postnr, poststed)
- ◆ butikkID  $\rightarrow$  butikknavn: Bryter BCNF
  - ◆ butikkID<sup>+</sup> = {butikkID, butikknavn, butikktype, adresse, postnr, poststed}
  - ◆ Dekomponerer til S<sub>21</sub>(butikkID, butikknavn, butikktype, adresse, postnr, poststed) og S<sub>22</sub>(produktID, butikkID) (på BCNF)

# 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

Kandiadtnøkler (fra forige uke):

- {butikkID, produktID},
- {butikkID, navn, kategori}

1. Finn normalformen til relasjonen
2. Dekomponer relasjonen til BCNF

## Dekomponering av

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

- ◆ postnr  $\rightarrow$  poststed: Bryter BCNF
  - ◆  $\text{postnr}^+ = \{\text{postnr}, \text{poststed}\}$  (på BCNF)
  - ◆ Dekomponerer til  $S_{211}(\text{postnr}, \text{poststed})$  og  $S_{212}(\text{butikkID}, \text{butikknavn}, \text{butikktype}, \text{adresse}, \text{postnr})$  (på BCNF)

# Oppgave 2 – Løsning

---

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

med FDene:

- ◆ produktID → navn
- ◆ produktID → kategori
- ◆ produktID → pris
- ◆ navn, kategori → produktID
- ◆ butikkID → butikknavn
- ◆ butikkID → butikktype
- ◆ butikkID → adresse
- ◆ butikkID → postnr
- ◆ postnr → poststed

Kandiadtøkler (fra forige uke):

{butikkID, produktID},  
{butikkID, navn, kategori}

1. Finn normalformen til relasjonen
2. Dekomponer relasjonen til BCNF

**Dekomponering blir alstå:**

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

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

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

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

## Oppgave 3 (Vanskelig, ikke pensum!)

---

Lag et skjema på BCNF som inneholder dataene for 2019 i "Fisketillatelser med fartøytilknytning og kvotestørrelser" fra Fiskeridirektoratet:

<https://fiskeridir.no/Tall-og-analyse/AApne-data/AApne-datasett/Fartoey-eier-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 "2019"
- ◆ 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 slik:

```
DROP SCHEMA IF EXISTS fiskeri;

BEGIN;
CREATE SCHEMA fiskeri;
CREATE TABLE fiskeri.kvoter(
    Datapr text,
    FartøyID text,
    Registreringsmerke text,
    Tillatelsekode text,
    Tillatelse text,
    TillatelseID text,
    TillatelseGjelderFraDato timestamp,
    TillatelseGjelderTilDato timestamp,
    Linjenummer int,
    Linjenummerbeskrivelse text,
    Kvotestørrelse text,
    KvotestrGjelderFraDato date,
    KvotestrGjelderTilDato date
);
COMMIT;
```

## Oppgave 3 – Løsning: Datainnlasting

---

- ◆ Åpner CSV-filen med f.eks. Vim, og utfører følgende search/replace:

```
:%s/\(\d\d\)\.\(\d\d\)\.\(\d\d\d\d\)/\3-\2-\1/g
```

for å få datoer på riktig format

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

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

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

## Oppgave 3 – Løsning: Bestemme FDer og dekomponering

---

FDer:

1. FartøyID → Registreringsmerke
2. Tillatelsekode → Tillatelse
3. TillatelseID → Datapr
4. TillatelseID → FartøyID
5. TillatelseID → Tillatelsekode
6. TillatelseID → TillatelseGjelderFraDato
7. TillatelseID → TillatelseGjelderTilDato
8. TillatelseID → KvotestrGjelderFraDato
9. TillatelseID → KvotestrGjelderTilDato
10. TillatelseID, Linjenummer → Linjenummerbeskrivelse
11. TillatelseID, Linjenummer → Kvotestørrelse

Kandidatnøkkel: {TillatelseID, Linjenummer}

## Oppgave 3 – Løsning: Dekomponering

---

```
BEGIN;  
  
-- FartøyID -> Registreringsmerke bryter med BCNF  
-- Tillukningen til FartøyID er {FartøyID, Registreringsmerke},  
-- altså får vi følgende (som ikke bryter med BCNF):  
CREATE TABLE fiskeri.Fartøy (  
    FartøyID text PRIMARY KEY,  
    Registreringsmerke text  
);
```

## Oppgave 3 – Løsning: Dekomponering

---

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

## Oppgave 3 – Løsning: Dekomponering

---

```
-- Har nå en tabell med alle attributter bortsett fra Registreringsmerke og Tillatelse
-- TillatelseID -> Datapr bryter med BCNF
-- Tillukningen til TillatelseID er {TillatelseID, Datapr, FartøyID, Tillatelsekode,
--                                     TillatelseGjelderFraDato, TillatelseGjelderTilDato,
--                                     KvotestrGjelderFraDato, KvotestrGjelderTilDato},
-- så får følgende (som ikke bryter BCNF):
CREATE TABLE fiskeri.Tillatelse (
    Datapr text,
    FartøyID text REFERENCES fiskeri.Fartøy(FartøyID),
    Tillatelsekode text REFERENCES fiskeri.TillatelseInfo(Tillatelsekode),
    TillatelseID text PRIMARY KEY,
    TillatelseGjelderFraDato timestamp,
    TillatelseGjelderTilDato timestamp,
    KvotestrGjelderFraDato date,
    KvotestrGjelderTilDato date
);
```

## Oppgave 3 – Løsning: Dekomponering

---

```
-- Står nå igjen med følgende tabell, som ikke bryter med BCNF:  
CREATE TABLE fiskeri.TillatelseDetaljer (  
    TillatelseID text REFERENCES fiskeri.Tillatelse(TillatelseID),  
    Linjenummer int,  
    Linjenummerbeskrivelse text,  
    Kvotestørrelse text,  
    CONSTRAINT td_pk PRIMARY KEY (TillatelseID, Linjenummer)  
);
```

## Oppgave 3 – Løsning: Migrering

---

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

```
INSERT INTO fiskeri.Fartøy
SELECT DISTINCT FartøyID, Registreringsmerke
FROM fiskeri.kvoter;
```

```
INSERT INTO fiskeri.TillatelseInfo
SELECT DISTINCT Tillatelsekode, Tillatelse
FROM fiskeri.kvoter;
```

```
INSERT INTO fiskeri.Tillatelse
SELECT DISTINCT Datapr, FartøyID, Tillatelsekode, TillatelseID,
    TillatelseGjelderFraDato, TillatelseGjelderTilDato,
    KvotestrGjelderFraDato, KvotestrGjelderTilDato
FROM fiskeri.kvoter;
```

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

```
COMMIT;
```