

IN2090

Funksjonelle avhengigheter, nøkler og normalformer



Gode og dårlige databaseskjemaer

- Godt skjema: Brnavn bestemmer alt om en student, Kurskode alt om et kurs!
- Dårlig skjema: Ingen (minste mengde) attributter bestemmer resten!

Brnavn	Navn	Etternavn	Adresse	Kurskode	Tittel	Beskrivelse	AntSP	Kara
evgenit	Evgenij	Thorstensen	Addr1	IN2090	Databaser	EnBeskr...	10	B
peternl	Petter	Nilsen	Addr2	IN2090	Databaser	EnBeskr...	10	A
evgenit	Evgenij	Thorstensen	Addr1	IN2080	Beregn...	Descr...	10	A
leifhka	Leif H.	Karlsen	Addr3	IN2090	Databaser	EnBeskr...	10	B
leifhka	Leif H.	Karlsen	Addr3	IN3110	Program...	EnBeskr2...	5	C

Student

Brnavn	Navn	Etternavn	Adresse
evgenit	Evgenij	Thorstensen	Addr1
peternl	Petter	Nilsen	Addr2
leifhka	Leif H.	Karlsen	Addr3

Kurs

Kurskode	Tittel	Beskrivelse	AntSP
IN2090	Databaser	EnBeskr...	10
IN2080	Beregn...	Descr...	10
IN3110	Program...	EnBeskr2...	5

Karakter

Brnavn	Kurskode	Kara
evgenit	IN2090	B
peternl	IN2090	A
evgenit	IN2080	B
leifhka	IN2090	B
leifhka	IN3110	C



Database design

- God database design handler om:
 - Data om entiteter i hver sin tabell.
 - Relasjoner mellom entiteter via referanser.
- Normalformer:
 - Det går an å formelt definere kriterier som gjør at anomalier ikke forekommer.
 - Et skjema som oppfyller kriteriene, sies å være på en normalform.
 - Vi skal se på normalformene 1NF, 2NF, 3NF, og BCNF.



Funksjonelle avhengigheter (FDer)

- Et attributt A er funksjonelt avhengig av en mengde attributter X hvis det bare kan finnes en verdi av A for hver mengde verdier av attributtene i X .
- Det skrives $X \rightarrow A$, og en slik formel kalles en funksjonell avhengighet (FD).
- Vi leser ofte pilen som "bestemmer", så $X \rightarrow Y$ leses enten
 - "X bestemmer Y" eller
 - "Y er funksjonelt avhengig av X"



FDers oppførsel

- Vi kan samle opp høyresider i FDer, og skrive $X \rightarrow A, B$ dersom vi både har $X \rightarrow A$ og $X \rightarrow B$.
- FDer er transitive: Hvis $X \rightarrow Y$ og $Y \rightarrow Z$, så har vi at $X \rightarrow Z$



EKSEMPEL

a)

Finn de funksjonelle avhengighetene til relasjonen: `Student(studentNr, land, populasjon)` som relaterer studenter til landet de studerer i, og hvor

1. `studentNr` er unikt for en student (altså attributten som kun er knyttet til en student, altså `studentNr` og `land`)
2. `populasjon` er populasjonen til landet med navn oppgitt i `land`

Løsningsforslag

a)

1. `studentNr` → `land`
2. `land` → `populasjon`



Nøkler

- En supernøkkel for en relasjon er enhver mengde attributter som sammen er unike for relasjonen
- En kandidatnøkkel er en minimal supernøkkel
- Dersom en mengde attributter er unike forekommer hver kombinasjon av disse kun i et tuppel, og bestemmer derfor de andre verdiene i tuplet
- Med andre ord, en nøkkel (enten super eller kandidat) er en mengde attributter som bestemmer de andre attributtene i relasjonen



Nøkler og FDer

- FDer sier hvilke attributter som bestemmer hvilke andre attributter. Altså, FDene sier hvilke supernøkler og kandidatnøkler vi har!
- For å sjekke om X er en supernøkkel, sjekk om alt er avhengig av X
- Altså, bruk FDene og finn alle attributter som er avhengige av X , de som er avhengige av disse igjen, osv.



Tillukning

- Tillukningen til X på en mengde F Der er mengden attributter som er funksjonelt avhengige av X
- Hvis $X \rightarrow A$, så er $A \in X^+$ sant
- Hvis $A \notin X^+$, så er ikke $X \rightarrow A$ sant



EKSEMPEL 1

b)

Gitt følgende relasjon: `Student(navn, personnr, fødselsdato, kurskode, fagkode, fagnummer, kursnavn)` som relaterer en student til kursene den tar, og FDene:

- `personnr → navn`
- `personnr → fødselsdato`
- `kurskode → kursnavn, fagkode, fagnummer`
- `fagkode, fagnummer → kurskode`

Finn tillukningen til 1. `personnr` 2. `{personnr, fagkode, fagnummer}`

Løsningsforslag

b)

1. `personnr+ = {personnr, navn, fødselsdato}`
2. `{personnr, fagkode, kurskode}+ = {personnr, fagkode, fagnummer, navn, fødselsdato, kurskode, kursnavn}`



EKSEMPEL 2

Gitt følgende relasjon

```
Bok(isbn, tittel, forfatter, forlagsnr, forlagsnavn, utgitt, sjanger, aldersgrense)
```

hvor vi har følgende Fder:

- isbn → tittel, forfatter, forlagsnr, sjanger
- forlagsnr → forlagsnavn
- utgitt, forfatter, forlagsnr → isbn
- sjanger → aldersgrense

a. Hva er tillukningen til forlagsnr ?

Løsningsforslag

a)

```
forlagsnr+ = {forlagsnr, forlagsnavn}
```



EKSEMPEL 3

Gitt følgende relasjon

```
Bok(isbn, tittel, forfatter, forlagsnr, forlagsnavn, utgitt, sjanger, aldersgrense)
```

hvor vi har følgende Fder:

- isbn → tittel, forfatter, forlagsnr, sjanger
- forlagsnr → forlagsnavn
- utgitt, forfatter, forlagsnr → isbn
- sjanger → aldersgrense

b. Hva er tillukningen til {forlagsnr, sjanger, forfatter}?

Løsningsforslag

b)

```
{forlagsnr, sjanger, forfatter}+ = {forlagsnr, sjanger, forfatter, forlagsnavn, aldersgrense}
```



Finne kandidatnøkler

- Vi må sjekke alle delmengder av attributter, nedenfra.
 - Men, følgende to regler hjelper oss:
 1. Hvis A ikke forekommer i noen høyreside, er A med i alle kandidatnøkler.
 2. Hvis A forekommer i minst en høyreside, men ingen venstresider, er A ikke del av noen kandidatnøkkel.
1. Så begynn med alle attributter som ikke forekommer på høyre side.
 2. Beregn tillukningen.
 3. Hvis alle attributter er med, sjekk minimalitet.
 4. Hvis ikke, utvid i tur og orden med ett og ett nytt attributt.



EKSEMPEL

Gitt følgende relasjon

```
Bok(isbn, tittel, forfatter, forlagsnr, forlagsnavn, utgitt, sjanger, aldersgrense)
```

hvor vi har følgende Fder:

- isbn → tittel, forfatter, forlagsnr, sjanger
- forlagsnr → forlagsnavn
- utgitt, forfatter, forlagsnr → isbn
- sjanger → aldersgrense

c. Hva er kandidatnøkklene til Bok ?



Løsningsforslag

c)

- Aldri på høyresider: `utgitt`
- Kun er på høyresider: `tittel, forlagsnavn, aldersgrense`
- Må forsøke å utvide med: `isbn, forlagsnr, forfatter, sjanger`

1. Forsøker først med kun `utgitt`: `utgitt+ = {utgitt}`, altså ikke kandidatnøkkel.
2. Utvider med `isbn`: `{isbn, utgitt}+ = {isbn, utgitt, tittel, forfatter, forlagsnr, sjanger, forlagsnavn, aldersgrense}`, altså kandidatnøkkel
3. Utvider med `forlagsnr`: `{utgitt, forlagsnr}+ = {utgitt, forlagsnr, forlagsnavn}`, altså ikke kandidatnøkkel
4. Utvider med `forfatter`: `{utgitt, forfatter}+ = {utgitt, forfatter}`, altså ikke kandidatnøkkel
5. Utvider med `sjanger`: `{utgitt, sjanger}+ = {utgitt, sjanger, aldersgrense}`, altså ikke kandidatnøkkel
6. Utvider med `forlagsnr` og `forfatter`:
`{utgitt, forlagsnr, forfatter}+ = {utgitt, forlagsnr, forfatter, forlagsnavn, isbn, tittel, sjanger, aldersgrense}`, altså en kandidatnøkkel
7. Utvider med `forlagsnr` og `sjanger`: `{utgitt, forlagsnr, sjanger}+ = {utgitt, forlagsnr, sjanger, forlagsnavn, aldersgrense}`, altså ikke kandidatnøkkel
8. Utvider med `forfatter` og `sjanger`: `{utgitt, forfatter, sjanger}+ = {utgitt, forfatter, sjanger, aldersgrense}`, altså ikke kandidatnøkkel

Alle større utvidelser vil kun gi supernøkler som ikke er minimale. Altså har vi følgende to kandidatnøkler: `{utgitt, isbn}` og `{utgitt, forlagsnr, forfatter}`.



Normalformer

- Normalformene vi skal se på danner et hierarki:

$$\text{BCNF} \subseteq \text{3NF} \subseteq \text{2NF} \subseteq \text{1NF}$$

- Det vil si:
Hvis et skjema oppfyller 3NF, oppfyller det også 2NF og 1NF



1NF

- En tabell er på 1NF hvis alle attributter er atomære
- Vi antar derfor at 1NF alltid er oppfylt



2NF

- En tabell oppfyller 2NF hvis den
 - oppfyller 1NF og
 - alle attributter A som ikke er nøkkelattributter, ikke er funksjonelt avhengige av en delmengde av en kandidatnøkkel
- En tabell **bryter** 2NF hvis det finnes et ikke-nøkkelattributt A som **er avhengig** av en delmengde av en kandidatnøkkel.



3NF

- En tabell oppfyller 3NF hvis
 - den oppfyller 2NF og
 - alle ikke-nøkkelattributter kun er avhengige av kandidatnøkler
- En tabell **bryter** 3NF hvis det finnes et ikke-nøkkelattributt som **er avhengig** av noe som **ikke** er en kandidatnøkkel.



BCNF

- Kort for Boyce-Codd normalform
- En tabell oppfyller BCNF hvis alle attributter kun er avhengige av en kandidatnøkkel
- Samme som 3NF, men unntaket for nøkkelattributter er fjernet
- Unntaket blir sjeldent brukt, og som regel er tabeller på 3NF også på BCNF



Normalformer i praksis

- Jo høyere normalform, jo færre anomalier
- Men jo høyere normalform, jo fler relasjoner i skjemaet
- BCNF fjerner (nesten) alle anomalier og er et godt kompromiss
- De fleste ordentlige skjemaene (Northwind, filmdb, osv.) er på BCNF

Jobb med ukesoppgaver/Innlevering 2

- Innlevering 2 (Enkel SQL)
[innlevering2.pdf \(uio.no\)](#)
- **Frist for Innlevering 2 (Enkel SQL): 12.Oktober kl 23.59!**
- Ukesoppgaver (uke 7: Modellering: FDer og normalformer)
[IN2090-ukesoppgaver: Uke 7 – Universitetet i Oslo \(uio.no\)](#)
- Utsettelse på oblig send mail til: camilldb@uio.no

