`IN2090-h21`: Obligatorisk innlevering 3

Martin Mihle Nygaard (`martimn`)

Oppgave 1 — Lage databaser

Jeg vedlegger SQL koden som egen fil, men inkluderer her og:

```
DROP TABLE IF EXISTS Tog;
DROP TABLE IF EXISTS TogTabell;
DROP TABLE IF EXISTS Plass;

CREATE TABLE Tog (
  togNr int PRIMARY KEY,
  startStasjon text NOT NULL,
  endeStasjon text NOT NULL,
  ankomstTid timestamp NOT NULL,
);

CREATE TABLE TogTabell (
  togNr int REFERENCES Tog(togNr),
  adgangsTid timestamp
  stasjon text NOT NULL,
  PRIMARY KEY (togNr, adgangsTid)
);

CREATE TABLE Plass (
  dato date
  togNr int REFERENCES Tog(togNr),
  vognNr int
  plassNr int
  vindu boolean NOT NULL,
  ledig boolean NOT NULL,
  PRIMARY KEY (dato, togNr, vognNr, plassNr)
);

-- Eneste skranker jeg putter er å ikke tillate nullverdier.
-- Datatypene bør være ganske selvforklarende: heltall på alt som
-- har et «nummer», `timestamp` på alle «tider» (klokkeslett og
-- dato), `date` det ene stedet oppgaven spør om bare dato, og
```

```
-- boolske verdier på vindusplass og ledighet.
```

Oppgave 2 — FDer og Normalformer

Kandidatnøklene til $\mathcal{R}$

Nøklene $\{C, F\}$ forekommer kun på venstre side av de funksjonelle avhengighettene, jeg vet da derfor at de nødvendigvis må være en del av alle kandidatnøkler. $\{G\}$, derimot forekommer kun på høyre side, og kan derfor ikke være inkludert i noen kandidatnøkler. Jeg finner tillukningen til $\{C, F\}$, og utvider den alfabetisk og rekursivt for å finne alle kandidatnøkler; jeg stryker hvis utvidelsen inneholder en annen kandidatnøkkel, mengden nøkler allerede er vurdert, eller det utvides med en allerede inkludert nøkkel.

- $\{C, F\}^+ = \{C, F\}$
  - $\{C, F, A\}^+ = \{C, F, A, B, D, E, G\} \rightarrow$ kandidatnøkkel
  - $\{C, F, B\}^+ = \{C, F, A, B, D, E, G\} \rightarrow$ kandidatnøkkel
  - $\cancel{\{C, F, C\}^+}$
  - $\{C, F, D\}^+ = \{C, F, D, G\}$
    - $\cancel{\{C, F, D, A\}^+}$
    - $\cancel{\{C, F, D, B\}^+}$
    - $\cancel{\{C, F, D, C\}^+}$
    - $\cancel{\{C, F, D, D\}^+}$
    - $\{C, F, D, E\}^+ = \{C, F, D, G, E, B, A\} \rightarrow$ kandidatnøkkel
    - $\cancel{\{C, F, D, F\}^+}$
    - $\cancel{\{C, F, D, G\}^+}$
  - $\{C, F, E\}^+ = \{C, F, E\}$
    - $\cancel{\{C, F, E, A\}^+}$
    - $\cancel{\{C, F, E, B\}^+}$
    - $\cancel{\{C, F, E, C\}^+}$
    - $\cancel{\{C, F, E, D\}^+}$
    - $\cancel{\{C, F, E, E\}^+}$
    - $\cancel{\{C, F, E, F\}^+}$
    - $\cancel{\{C, F, E, G\}^+}$
  - $\cancel{\{C, F, F\}^+}$
  - $\cancel{\{C, F, G\}^+}$

Sorry, litt mye som skjer her, men oppsummert: kandidatnøklene blir $\{C, F, A\}$, $\{C, F, B\}$ og $\{C, F, D, E\}$.

Høyeste normalform til $\mathcal{R}$

Jeg følger algoritmen fra forelesning. Jeg har allerede funnet kandidatnøklene: $\{C, F, A\}$, $\{C, F, B\}$ og $\{C, F, D, E\}$. Jeg vurderer de funksjonelle avhengighetene etter tur:

- $CDE \to B$: brudd på BCNF, siden CDE ikke er en supernøkkel.
  - $CDE \to B$: på 3NF, siden B er et nøkkelattributt.
- $AF \to B$: brudd på BCNF, siden AF ikke er en supernøkkel.
  - $AF \to B$: på 3NF, siden B er er et nøkkelattributt.
- $B \to A$: brudd på BCNF, siden B ikke er en supernøkkel.
  - $B \to A$: på 3NF, siden A er et nøkkelattributt.
- $BCF \to DE$: på BCNF, siden BCF er en supernøkkel.
- $D \to G$ brudd på BCNF, siden D ikke er en supernøkkel.
  - $D \to G$: brudd på 3NF, siden G ikke er et nøkkelattributt.
    - $D \to G$: på 2NF, siden D er en del av en kandidatnøkkel.

Høyeste normalformen som $\mathcal{R}$ tilfredsstiller er 2NF.

(c) — Tapsfri dekomponering[1]

Bruker algoritmen fra forelesning:

> Tapsfri dekomponering av $R(X)$ med funksjonelle avhengigheter F:
>
> 1. Beregn nøklene til R
> 2. For hver funksjonell avhengighet $Y \to A \in F$, hvis den funksjonelle avhengigheten er et brudd på BCNF:
>   i. beregn Y^+,
>   ii. og dekomponer R til $S_1(Y^+)$ og $S_2(Y, X/Y^+)$.
> 3. Fortsett rekursivt (over S_1 og S_2) til ingen brudd på BCNF

Jeg navngir de forskjellige funksjonelle avhengighetene i tabellen under. Deretter utfører jeg algoritmen på dem etter tur.

I	$CDE \to B$
II	$AF \to B$
III	$B \to A$
IV	$BCF \to DE$
V	$D \to G$

$CDE \to B$ Bryter BCNF, siden CDE ikke er en supernøkkel. Jeg dekomponerer til $S_1(CDE^+) = S_1(ABCDEG)$ og $S_2(CDE, ABCDEFG/CDE^+) = S_2(CDEF)$. S_2 har ingen funksjonelle avhengigheter, men S_1 har III og V som bryter med BCNF.

Jeg dekomponerer først S_1 til $S_{11}(B^+) = S_{11}(AB)$ og $S_{12}(B, ABCDEG/B^+) = S_{12}(CDEG)$.

- $S_{11}(AB)$ har kun III, men denne er på BCNF.

[1] Jeg er usikker på om jeg har tenkt riktig i denne oppgaven. Den ble veldig innviklet. Håper det som følger en noenlunde forståelig.

- S_{12} har V, som bryter med BCNF. Jeg dekomponerer S_{12} til $S_{121}(D^+) = S_{121}(DG)$ og $S_{122}(D, CDEG/D^+) = S_{122}(CDE)$. Verken S_{121} eller S_{122} har funksjonelle avhengigheter som bryter BCNF.

Jeg dekomponerer også S_1 til $T_{11}(D^+) = T_{11}(DG) = S_{121}$ og $T_{12}(D, ABCDEG/D^+) = T_{12}(ABCDE)$.

- $T_{11}(DG)$ er lik S_{121} og kan ignoreres.
- $T_{12}(ABCDE)$ har I, som ikke bryter, og III, som bryter med BCNF. Jeg dekomponerer T_{12} til $T_{121}(B^+) = T_{121}(AB) = S_{11}$ og $T_{122}(B, ABCDE/B^+) = T_{122}(BCDE)$. Jeg kan ignorere $T_{121} = S_{11}$, mens T_{122} har ingen funksjonelle avhengigheter som bryter BCNF.

$\boldsymbol{AF \to B}$ Bryter BCNF, siden AF ikke er en supernøkkel. Jeg dekomponerer $\mathcal{R}$ til $U_1(AF^+) = U_1(ABF)$ og $U_2(AF, ABCDEFG/AF^+) = U_2(CDEG) = S_{12}$.

- $U_1(ABF)$: har funksjonelle avhengigheter II og III. Siden III bryter BCNF, dekomponerer jeg U_1 til $U_{11}(B^+) = U_{11}(AB) = S_{11}$ og $U_{12}(B, ABF/B^+) = U_{12}(BF)$. Begge er på BCNF.
- $U_2 = S_{12}$ så denne er allerede tatt hånd om.

$\boldsymbol{B \to A}$ Bryter BCNF. Jeg dekomponerer $\mathcal{R}$ til $V_1(B^+) = V_1(AB) = S_{11}$ og $V_2(B, ABCDEFG/B^+) = V_2(BCDEFG)$.

- $V_1(AB) = S_{11}$ er allerede vurdert.
- $V_2(BCDEFG)$: har FDer I og V som bryter BCNF (og IV som ikke bryter). Jeg dekomponerer V_2 til $V_{21}(CDE^+) = V_{21}(BCDEG)$ og $V_{22}(CDE, BCDEFG/CDE^+) = V_{22}(CDEF) = S_2$.
  - V_{21} har kun V som bryter med BCNF. Jeg dekomponerer derfor til V_{21} til $V_{211}(D^+) = V_{211}(DG) = S_{121}$ og $V_{212}(D, BCDEG/D^+) = V_{212}(BCDE) = T_{122}$, som begge alt er vurdert.
  - $V_{22} = S_2$ og er alt rekursert over.

$\boldsymbol{D \to G}$ Bryter BCNF. Jeg dekomponerer derfor til $W_1(D^+) = W_1(DG) = S_{121}$ og $W_2(D, ABCDEFG/D^+) = W_2(ABCDEF)$.

- $W_1 = S_{121}$ er alt vurdert.
- W_2 har FDer I, II, III og IV som alle bryter med BCNF.
  - Jeg dekomponerer W_2 til $W_{211}(CDE^+) = W_{211}(ABCDE) = T_{12}$ og $W_{212}(CDE, ABCDEF/CDE^+) = W_{212}(CDEF) = S_2$. Begge relasjonene er allerede vurdert.
  - Jeg dekomponerer W_2 til $W_{221}(AF^+) = W_{221}(ABF) = U_1$ og $W_{222}(AF, ABCDEF/AF^+) = W_{221}(ACDEF)$. Hvor $W_{221} = U_1$ allerede er vurdert, W_{221} har ingen funksjonelle avhengigheter.

Oppsummering Dette ble veldig rotete, men over dekomponerer jeg $\mathcal{R}(ABCDEFG)$ til følgende relasjoner (med litt vilkårlige navn):

- $S_{11}(AB)$
- $S_{122}(CDE)$
- $S_{121}(DG)$
- $T_{122}(BCDE)$
- $U_{12}(BF)$
- $W_{221}(ACDEF)$