

Kogebog

Jeg tager ikke ansvar for noget angående jeres eksamen. Nedenstående afspejlet alene min forståelse indtil videre, og jeg har ikke læst korrektur. Jeg tager gerne imod kommentarer og tænker, at vi i fællesskab kan lave os en kogebog.

Forklaringerne til loss less join og dependency preservation kommer ikke fra bogen, men fra

<https://www.youtube.com/watch?v=TykMe1A2u6U>

<https://www.youtube.com/watch?v=9PZzyMhQVw>

Design teori

Eksempel

recipes(dish_name, cuisine_tradition, country_of_origin, is_vegetarian, is_vegan, is_dairy_free, is_egg_free, ingredient_name, ingredient_quantity)

Bemærk venligst, at jeg for nemheds skyld har nummereret de funktionelle afhængigheder.

FD1 dish_name → country_of_origin, is_vegetarian, is_vegan, is_dairy_free, is_egg_free

FD2 dish_name, ingredient_name → ingredient_quantity

FD3 country_of_origin → cuisine_tradition

FD4 is_vegetarian, is_dairy_free, is_egg_free → is_vegan

Normalisering

En relation er i BCNF hvis

- 1) Den er 3NF
- 2) Der er en superkey på venstresiden af ALLE FD'ere

Sagt med andre ord. Den minimale nøgle/nøgle(r), der i kraft af de funktionelle afhængigheder giver os alle attributterne, skal være til stede på ALLE venstresiderne. En superkey kan godt indeholde flere elementer derudover, men den/det minimale sæt nøgle(r) skal være til stede.

Brud på BCNF:

non-prime attribute → prime attribute

En relation er i 3NF

- 1) Hvis den er i 2NF

2) Der ikke er nogen transitiv afhængighed

Transitive dependency

non-prime attribute \rightarrow non-prime attribute

En relation er i 2NF

1) Hvis den er i 1NF

2) Der ikke er nogen partiel afhængighed

Partial dependency

part of primary key \rightarrow non-prime attribute

En relation er i 1NF

1) Hvis alle værdier er enkeltstående (atomic).

2) Hver kolonne kun indeholder værdier af samme type.

3) Hver kolonne har et unikt navn.

4) Rækkefølgen af kolonnerne hhv. rækkerne ikke betyder noget.

Bestem den minimale superkey

Start med venstresiden af en af de funktionelle afhængigheder og beregn dens "closure" således:

Sæt venstresiden lig med X. Tilføj alle elementer, som de funktionelle afhængigheder tillader

{dish_name}⁺ beregnes:

X = {dish_name} Første og tredje FD resulterer i

X = {dish_name, country_of_origin, is_vegetarian, is_vegan, is_dairy_free, is_egg_free, cuisine_tradition}

Resultatet skrives

{dish_name}⁺ = {dish_name, country_of_origin, is_vegetarian, is_vegan, is_dairy_free, is_egg_free, cuisine_tradition}

Dette udelader ingredient_name og ingredient_quantity, så vi beregner {dish_name, ingredient_name}⁺

X = {dish_name, ingredient_name} FD1, FD3 og FD2 resulterer i

\newline X = {dish_name, country_of_origin, is_vegetarian, is_vegan, is_dairy_free, is_egg_free, cuisine_tradition, ingredient_name, ingredient_quantity}

$\{dish_name, ingredient_name\}^+ = \{dish_name, country_of_origin, is_vegetarian, is_vegan, is_dairy_free, is_egg_free, cuisine_tradition, ingredient_name, ingredient_quantity\}$ includes all elements of the relation recipes.

Den minimale superkey er derfor dish_name og ingredient_name.

Dekomposition

Undersøg om ALLE FD'erne har en superkey på venstresiden. Hvis det ikke er tilfældet, skal relationen dekomponeres. Til det formål benytter vi de FD'ere som ikke er i overensstemmelse med BCNF (BCNF violation).

I det følgende er megachartet betegnet R1. Det skal splittes i R2 og R3

Vælg en FD som ikke er i overensstemmelse med BCNF.

R2 skal bestå af indholdet både af venstre- og højresiden af FD.

R3 skal bestå af FD'ens venstreside plus de elementer i R1, som ikke er med i R2. Med andre ord venstresiden og det, som blev overs af R1.

Sammenlign R2 med FD'erne. Hvis indholdet af R2 nøjagtigt svarer til det samlede indhold (både venstre- og højresiden) af en FD, så er det subschema i orden.

Hvis R3 stadig ligner en rodebutik gentages processen. Vi er i mål når det sidste subschema enten svarer til indholdet af en FD, eller indholder den minimale superkey plus en højreside, som ikke kunne komme med i de andre subschemaer.

Hvis ingen af FD'erne er i overensstemmelse med BCNF betyder rækkefølgen ikke så meget. (Eksemplet med chart).

Hvis en af FD'erne rent faktisk har en superkey på venstresiden, bør man vælge rækkefølgen således, at indholdet af det sidste subschema svarer til indholdet (både venstre og højresiden) af den sidste FD, som man benytter. Med andre ord, man vælger den sidste FD, så venstresiden tilsammen med resten (det der bliver til overs af det skema, som skal nedbrydes) giver indholdet af den funktionelle afhængighed, som rent faktisk havde en superkey på venstresiden. (Eksemplet med recipes).

FD1 dish_name → country_of_origin, is_vegetarian, is_vegan, is_dairy_free, is_egg_free

FD2 dish_name, ingredient_name → ingredient_quantity

FD3 country_of_origin → cuisine_tradition

FD4 is_vegetarian, is_dairy_free, is_egg_free → is_vegan

Relationen recipes er ikke i BCNF, fordi første, tredje og fjerde FD ikke har en superkey på venstresiden. Undtagelsen er FD2, hvilket betyder, at vi skal holde tungen lige i munden, når vi vælger rækkefølgen i

hvilken vi benytter de FD'ere som ikke er i BCNF. Det kan være nødvendigt at lege lidt med rækkefølgen før det hele falder på plads.

Den er heller ikke i 3NF på grund af FD3 og FD4.

dish_name → country_of_origin → cuisine_tradition

dish_name → is_vegetarian, is_dairy_free, is_egg_free → is_vegan

R1(dish_name, cuisine_tradition, country_of_origin, is_vegetarian, is_vegan, is_dairy_free, is_egg_free, ingredient_name, ingredient_quantity)

R2: country_of_origin, cuisine_tradition (Indholdet af FD3)

R3: country_of_origin, dish_name, is_vegetarian, is_vegan, is_dairy_free, is_egg_free, ingredient_name, ingredient_quantity (Resten)

R2 er nu i overensstemmelse med FD3, men R3 skal brydes om til R4 og R5. Målet er, at R5 kommer til at svare til FD2. Man skal vælge sin FD, så resten kommer til at bestå af ingredient_name og ingredient_quantity, som skal fusioneres med dish_name. Heldigvis udgør dish_name venstresiden af FD1, hvorfor FD1 bliver udgangspunktet for den sidste dekomposition:

R4: dish_name, country_of_origin, is_vegetarian, is_vegan, is_dairy_free, is_egg_free

R5: dish_name, ingredient_name, ingredient_quantity

Relationen recipes er nu brudt om i følgende underordnede relationer.

R2: country_of_origin, cuisine_tradition

R4: dish_name, country_of_origin, is_vegetarian, is_vegan, is_dairy_free, is_egg_free

R5: dish_name, ingredient_name, ingredient_quantity

Loss less join (Forklaringen er lidt anderledes i bogen)

Lav et skema med en kolonne for hvert af attributterne i den oprindelige megarelation og en række for hvert subskema/underordnet relation:

Sæt kryds i hvert felt, hvor et attribut optræder i det pågældende subskema. Krydserne repræsenterer (distinguished variables).

| | Dish_name | Cuisine_tradition | Country_of_origin | Is_vegetarian | Is_vegan | Is_dairy_free | Is_egg_free | Ingredient_name | Ingredient_quantity |
|----|-----------|-------------------|-------------------|---------------|----------|---------------|-------------|-----------------|---------------------|
| R2 | | a | a | | | | | | |
| R4 | a | | a | a | a | a | a | | |
| R5 | a | | | | | | | a | A |

Skemaet sammenholdes nu med de funktionelle afhængigheder, for at se om vi må sætte flere krydser. Hvis vi kan få en hel række på langs, betyder det, at dekompositionen er loss less. Her kan det være en hjælp at bryde de funktionelle afhængigheder med to eller flere attributter på højresiden om til simple funktionelle afhængigheder med kun et element på højresiden.

dish_name → country_of_origin

dish_name → is_vegetarian

dish_name → is_vegan

dish_name → is_dairy_free

dish_name → is_egg_free

dish_name, ingredient_name → ingredient_quantity

country_of_origin → cuisine_tradition

is_vegetarian, is_dairy_free, is_egg_free → is_vegan

Nu skal vi finde tilfælde, som lever op til alle tre betingelser:

- 1) For en given funktionel afhængighed find 2 eller flere rækker som har attributter svarende til venstresiden.
- 2) Find blandt disse rækker mindst én, som også har den funktionelle afhængigheds højreside.
- 3) Find blandt disse rækker (altså dem, som overholder 1) mindst én, som mangler den funktionelle afhængigheds højreside

Hvis alle tre betingelser er opfyldt, må man sætte kryds på de tomme pladser.

Gentag processen for alle funktionelle afhængigheder.

Nu har du enten

- a) en række med krydser. Tillykke, din dekomposition er loss less!
- b) række med én eller flere tomme pladser

Hvis b, gentag da processen funktionel afhængighed for funktionel afhængighed, indtil du enten har en hel række, eller har været hele sættet af funktionelle afhængigheder igennem uden at kunne sætte et kryds. I så fald er dekompositionen lossy.

Eksempel: R4 og R5 er fælles om dish_name, så betingelse 1 er opfyldt.

R4 indeholder FD1's højreside, men det gør R5 ikke, så betingelse 2 og 3 er også opfyldt. Vi må sætte krydser (markeret med grønt).

Princip 1 kan ikke opfyldes hvad angår FD2, fordi kun R5 både har dish_name og ingredient_quantity

FD4 opfylder det første princip, fordi alle tre relations har en "distinguished variable" for country_of_origin.

FD4 opfylder også det andet og tredje princip, fordi R2 har cuisine_tradition, hvorimod R4 og R5 ikke har. Derfor må vi sætte kryds ud for R4 og R5. (markeret med rødt). Bingo! Nu er der en hel række ud for R5, hvilket betyder, at dekompositionen er loss less.

| | Dish_name | Cuisine_tradition | Country_of_origin | Is_vegetarian | Is_vegan | Is_dairy_free | Is_egg_free | Ingredient_name | Ingredient_quantity |
|----|-----------|-------------------|-------------------|---------------|----------|---------------|-------------|-----------------|---------------------|
| R2 | | a | a | | | | | | |
| R4 | a | a | a | a | a | a | a | | |
| R5 | a | a | a | a | a | a | a | a | a |

Opretholdelse af de funktionelle afhængigheder

Undersøg om ALLE funktionelle afhængigheder er bevaret i subskemaerne (R1, R2,...Rn).

Hvis bare én ikke er bevaret, så dur det ikke.

Den enkle metode:

Hvis både højre- og venstresiden af den funktionelle afhængighed er indeholdt i et af underskemaet, er den pågældende funktionelle afhængighed bevaret. Kontroller at alle funktionelle afhængigheder er bevaret.

FD1 dish_name → country_of_origin, is_vegetarian, is_vegan, is_dairy_free, is_egg_free er indeholdt i R4

FD2 dish_name, ingredient_name → ingredient_quantity er indeholdt i R5.

FD3 country_of_origin → cuisine_tradition er indeholdt i R2.

FD4 is_vegetarian, is_dairy_free, is_egg_free → is_vegan er indeholdt i R4.

Den hårde måde:

- 1) Vælg en funktionel afhængighed.
 - 2) Sæt venstresiden lig med Z.
 - 3) Bestem "closure" af fælles mængden af Z og R1
 - 4) Bestem fællesmængden af $\{Z \text{ fælles } R1\}^+$ og R1
 - 5) Hvis der er nogen elementer i fællesmængden af "closure" og R1, som ikke allerede er indeholdt i Z, tilføjes disse til Z.
 - 6) Hvis højresiden af den funktionelle afhængighed nu er kommet med i Z, er den funktionelle afhængighed bevaret, og vi kan gå videre til den næste funktionelle afhængighed.
- Hvis højresiden forsat mangler gentager vi processen med R2 og så fremdeles. Husk blot at det er Z, som den ser ud efter femte trin, som man arbejder videre med.
- Hvis man når alle subskemaer igennem uden at få højre siden med, men Z nu indeholder elementer, som ikke var der, da vi startede processen i trin to, så starter man processen forfra med subskema 1.
- Sådan starter man forfra, hver gang Z har fået nye elementer tilføjet indtil højresiden kommer med.
- Hvis man når alle subskemaer igennem fra R1 til Rn uden at Z har fået tilføjet noget nyt er dekompositionen ikke "dependency preserving".

Eksempel

Megarelation: <chart(patientid, patientname, doctorid, doctorname, doctorspecialty, condition, unittype, bednumber, illness)

Funktionelle afhængigheder for nemheds skyld nummereret:

FD1 patientid \rightarrow patientname, condition, bednumber, illness

FD2 doctorid \rightarrow doctorname, doctorspecialty

FD3 condition \rightarrow unittype

FD4 bednumber \rightarrow unittype

Den minimale nøgle

$\{\text{patientid}\}^+$

$X = \{\text{patientid}\}$ FD1, FD3 og FD4 tilføjer

$X = \{\text{patientid, patientname, condition, bednumber, illness, unittype}\}$

$\{\text{patientid}\}^+ = \{\text{patientid, patientname, condition, bednumber, illness, unittype}\}$

Dette udelader doctorid, doctorname og doctorspecialty. Derfor tilføjes doctorid

$\{\text{patientid, doctorid}\}^+$

$X = \{\text{patientid}, \text{doctorid}\}$ FD1, FD2, FD3 og FD4 tilføjer

$\{\text{patientid}, \text{doctorid}\}^+ = \{\text{patientid}, \text{patientname}, \text{condition}, \text{bednumber}, \text{illness}, \text{unitttype}, \text{doctorid}, \text{doctorname}, \text{doctorspecialty}\}$ Nu er hele indholdet af megarelationen med I lukket, hvorfor den minimale superkey er $\{\text{patientid}, \text{doctorid}\}$

Pointen med denne opgave er, at rækkefølgen med hvilken man anvender de funktionelle afhængigheder til at dekompositionere er afgørende for om de funktionelle afhængigheder bevares.

I dette eksempel er ingen af de funktionelle afhængigheder i overensstemmelse med BCNF, fordi ingen har en superkey på venstresiden.

Vi starter bagfra med $\text{bednumber} \rightarrow \text{unitttype}$

$R1(\text{patientid}, \text{patientname}, \text{doctorid}, \text{doctorname}, \text{doctorspecialty}, \text{condition}, \text{unitttype}, \text{bednumber}, \text{illness})$

$R2(\text{bednumber}, \text{unitttype})$ Det vil sige både højre- og venstresiden.

$R3(\text{bednumber}, \text{patientid}, \text{patientname}, \text{doctorid}, \text{doctorname}, \text{doctorspecialty}, \text{condition}, \text{illness})$
Venstresiden plus $R1-R2$ (dvs. venstresiden plus resten).

Indholdet af $R2$ svarer til FD4, så den er i orden, men $R3$ skal ombrydes. Vi tager udgangspunkt i FD3 $\text{condition} \rightarrow \text{unitttype}$

$R4(\text{condition}, \text{unitttype})$

$R5(\text{condition}, \text{bednumber}, \text{patientid}, \text{patientname}, \text{doctorid}, \text{doctorname}, \text{doctorspecialty}, \text{illness})$

$R4$ modsvarer $\text{condition} \rightarrow \text{unitttype}$, så den er i orden. $R5 = \text{venstresiden plus } R3 \text{ minus } R4$ er det samme, som indholdet af $R3$, så den skal stadig dekomponeres. Denne gang bruger vi FD2 : $\text{doctorid} \rightarrow \text{doctorname}, \text{doctorspecialty}$

$R6(\text{doctorid}, \text{doctorname}, \text{doctorspecialty})$

$R7(\text{doctorid}, \text{condition}, \text{bednumber}, \text{patientid}, \text{patientname}, \text{illness})$ Venstresiden plus $R5$ minus $R6$.

$R6$ modsvarer FD2 så den er ok.

Ved hjælp af FD1 dekomponerer vi en sidste gang

$R8(\text{patientid}, \text{patientname}, \text{condition}, \text{bednumber}, \text{illness})$ som modsvarer FD1

$R9(\text{patientid}, \text{doctorid})$

Megarelationen chart kan altså dekomponeres til:

$R2(\text{bednumber}, \text{unitttype})$, $R4(\text{condition}, \text{unitttype})$, $R6(\text{doctorid}, \text{doctorname}, \text{doctorspecialty})$, $R8(\text{patientid}, \text{patientname}, \text{condition}, \text{bednumber}, \text{illness})$ og $R9(\text{patientid}, \text{doctorid})$

Er dekompositionen loss less? Sorte a'er markerer attributterne. (distinguished variables).

| | Patientid | patientname | doctorid | doctorname | Doctorspecialty | condition | Unittype | bednummer | illness |
|----|-----------|-------------|----------|------------|-----------------|-----------|----------|-----------|---------|
| R2 | | | | | | | a | a | |
| R4 | | | | | | a | a | | |
| R6 | | | a | a | a | | | | |
| R8 | a | a | | | | a | a | a | a |
| R9 | a | a | a | a | a | a | a | a | a |

R8 og R9 er fælles om patientid, som udgør venstresiden i FD1. R8 har højresiden patientname, condition, bednummer, illness, som R9 mangler. Disse tilføjes som til R9 (markeret med rødt).

R6 og R9 er fælles om doctorid, som udgør venstresiden i FD2, men kun R6 har højresiden. Denne tilføjes til R9. (Markeret med grønt).

R4, R6 og R9 er fælles om condition, som udgør venstresiden i FD3, men kun R4 har højresiden unittype, den kan nu tilføjes til R6 og R9. (med lilla). Bingo dekompositionen er loss less.

Er de funktionelle afhængigheder bevaret?

FD1 patientid → patientname, condition, bednummer, illness er indeholdt i R8.

FD2 doctorid → doctorname, doctorspecialty er indeholdt i R6.

FD3 condition → unittype er indeholdt i R4

FD4 bednummer → unittype er indeholdt i R2.

Jeg måtte rode lidt med rækkefølgen med hvilken jeg benyttede de funktionelle afhængigheder. DS side 98 til 101 beskriver, hvordan man kan slække på kravene til BCNF og dekomponere med udgangspunkt i 3NF. Det vil sige, at man tager udgangspunkt i funktionelle afhængigheder, som ikke lever op til 3NF og benytter dem i stedet.