

Relasjonsdatabaseteori

Nøkler, funksjonelle avhengigheter og normalformer

Lana Vu anhlv@ifi.uio.no

Hva kjennetegner god relasjonsdatabasedesign?

- Relasjonene samler beslektet informasjon
- Så lite dobbeltlagring som mulig
- Korrekt totalinformasjon kan gjenskapes nøyaktig ved join

Relasjonene samler beslektet informasjon

- Tekstlig nærhet gjenspeiler logisk nærhet (Med tekstlig nærhet menes her samlokalisering i en relasjon)
- Brudd på dette prinsippet har en tendens til å påtvinge duplisering av data og dermed forårsake oppdateringsanomalier

Oppdateringsanomalier

Innsettingsanomalier

- Opprettholde konsistente verdier
- Håndtere sekundær informasjon
- Håndtere nil i kandidat- og fremmednøkler

Slettingsanomalier

- Unngå tap av sekundær informasjon

Modifiseringsanomalier

- Opprettholde konsistente verdier
- Oppdatere sekundær informasjon

Så lite dobbeltlagring som mulig

- Plassbehovet minimaliseres
- Oppdatering forenkles

Hvordan unngå dobbeltlagring?

- Splitt (dekomponer) relasjonene slik at dobbeltlagring blir borte!

Integritetsregler

- Integritetsregler begrenser mengden av lovlige instanser for et databaseskjema.
- Primærnøkler uttrykker én type integritetsregler.
- Primærnøkler er spesialtilfeller av kandidatnøkler.
- Kandidatnøkler er spesialtilfeller av funksjonelle avhengigheter.
- (I tillegg finnes andre typer integritetsregler)

Nøkler

- X er en **supernøkkel** i R : $X \subseteq R$, og ingen instans av R får inneholde to forskjellige tupler t_1 og t_2 hvor $t_1[X] = t_2[X]$.
- X er en **kandidatnøkkel** i R : X er en supernøkkel i R , og for alle A i X er $X-A$ ikke en supernøkkel i R . (Dvs. X er en minimal supernøkkel.)
- X er en **primærnøkkel** i R : X er en spesielt utpekt kandidatnøkkel i R .

Nøkkelattributt

- Et nøkkelattributt er et attributt som er med i en kandidatnøkkel.
- Et ikke-nøkkelattributt er et attributt som ikke er med i noen kandidatnøkkel.

Eksempel 1

Student(fnr, id, navn, adresse)

Primærnøkke: fnr

Kandidatnøkkler: id

Supernøkkler: 12 stk

{fnr}, {fnr,id}, {fnr,navn}, {fnr,adresse}, {fnr, navn, adresse}, {fnr,id,navn}, {fnr,id,adresse},
{fnr,id,navn,adresse}, {id}, {id,navn}, {id,adresse}, {id,navn,adresse}

Nøkkeattributter: fnr, id

Ikke-nøkkeattributter: navn, adresse

Eksempel 2

Filmgenre(filmid, genre, tittel)

Primærnøkkel: {filmid, genre}

Supernøkler: {filmid, genre}, {filmid, genre, tittel}

Nøkkelattributter: filmid, genre

Ikke-nøkkelattributter: tittel

Funksjonelle avhengigheter

Funksjonelle avhengigheter

- Gitt en relasjon R og integritetsregler for R , og gitt $X, Y \subseteq R$.
 - Y er **funksjonelt avhengig av** X hvis vi for enhver lovlig instans av R har at hvis instansen inneholder to tupler t_1 og t_2 hvor $t_1[X] = t_2[X]$, så må $t_1[Y] = t_2[Y]$.
 - I så fall skriver vi **$X \rightarrow Y$** .
- Ofte snakker vi for korthets skyld om «FDen $X \rightarrow Y$ » (der **FD** står for Functional Dependency)
- Vi sier at « Y følger av X », eller at « X bestemmer Y »
- Integritetsregelen $X \rightarrow A_1 A_2 \dots A_k$ kan alternativt representeres ved k FDer $X \rightarrow A_1$, $X \rightarrow A_2$, ..., $X \rightarrow A_k$ (hvor høyresidene består av bare ett attributt).

Funksjonell avhengighet og kandidatnøkler

- Merk at hvis X er en supernøkkel, så holder $X \rightarrow Y$ for enhver Y .
 - Så hvis X er en primærnøkkel, eller mer generelt en kandidatnøkkel, holder $X \rightarrow Y$ for enhver Y .
- Omvendt: Hvis $X \rightarrow Y$ for enhver Y , så er X en supernøkkel.
- Spesielt betyr dette at hvis vi angir at $R(A_1, A_2, \dots, A_n)$ har en kandidatnøkkel X , betyr det at R har FDen $X \rightarrow A_1 A_2 \dots A_n$.
 - Hvis $R(A_1, A_2, \dots, A_n)$ har en primærnøkkel X , betyr det at R har FDen $X \rightarrow A_1 A_2 \dots A_n$.

Funksjonelle avhengigheter

oppsummert

Enkel def: Et attributt er «avhengig»/bestemmes av et annet attributt/-er

Eksempel:

Student(fnr, id, navn, adresse, postnr, poststed)

Funksjonelle avhengigheter (FDer):

Fnr -> id, navn, adresse, postnr, poststed

Id -> fnr, navn, adresse, postnr, poststed

Kan skrives på denne måten:

Fnr -> navn

Fnr -> adresse

Fnr -> postnr

Fnr -> poststed

Fnr -> id

FD oppsummering forts.

Merk: En FD trenger nødvendigvis ikke å være en nøkkel!

Eksempel 1:

Student(fnr, id, navn, adresse, postnr, poststed)

Ny FD:

Et postnr bestemmer ett poststed

Postnr -> poststed.

Eksempel

Ordre(ordrenr, kundenr, kundenavn, antall, sum, mva)

- Ordrenr er unikt
- Sum bestemmer mva
- Kundenr bestemmer kundenavn

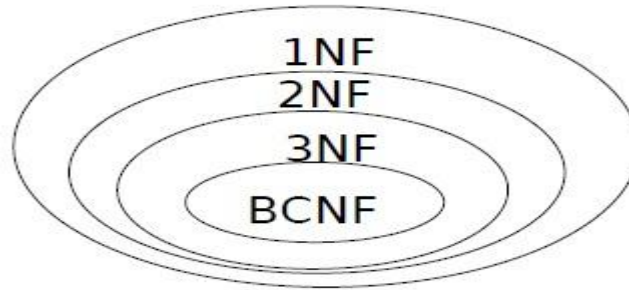
Finn de funksjonelle avhengighetene og nøklene for relasjonen Ordre.

Normalformer

Normalformer

- **Problem:** Hvordan vurdere objektivt om en samling relasjoner er god/dårlig?
- **Normalformer** er et uttrykk for hvor godt vi har lykket i en dekomposisjon
- At et skjema er på en normalform, sikrer at visse typer dobbeltlagring ikke forekommer
- Jo høyere normalform, jo mindre dobbeltlagring
- Vi kan bruke funksjonelle avhengigheter (FDer) til å sjekke hvilken normalform en relasjon er på
- Settet med FDer kan ha forskjellige normalformer, og relasjonen vil totalt være på den laveste normalformen fra settet med FDer.

Hvilke normalformer har vi



Ulike normalformer

Gitt en relasjon R, med et sett av FDer på formen $X \rightarrow A$, der X og A er et sett av attributter.

1NF:

- Bare atomære verdier/attributter

2NF:

- X er en supernøkkel i R
- A er et nøkkelattributt
- X er ikke en delmengde av noen nøkler i R

3NF:

- X er en supernøkkel i R
- A er et nøkkelattributt

BCNF(Boyce-Codd):

- X er en supernøkkel i R

Eksempel 1

Timeliste(ansattnr, uke, år, navn, timer)

FDer:

Ansattnr, uke, år \rightarrow navn, timer (BCNF)

Ansattnr \rightarrow navn (1NF)

Hvilken normalform er relasjonen på?

Eksempel 2

Ordre(ordrenr, kundenr, kundenavn, antall, sum, mva)

FDer:

ordrenr -> kundenr, kundenavn, antall, sum, mva (BCNF)

Kundenr -> kundenavn (2NF)

Sum -> mva (2NF)

Hvilken normalform er relasjonen på?

Eksempel 3

Student_Emne (id, emnekode, bnavn, karakter)

FDer:

Id, emnekode -> karakter (BCNF)

bnavn, emnekode -> karakter (BCNF)

Id -> bnavn (3NF)

Bnavn -> id (3NF)

Hvilken normalform er relasjonen på?

Eksempel 4

Student(fnr, id, navn, adresse)

FDer:

Fnr -> id, navn, adresse (BCNF)

Id -> fnr, navn, adresse (BCNF)

Hva gjør vi videre?

- Dekomposisjon til høyere normalform

Eksempel:

Fra 3NF til BCNF

Student_Emne(id, emnekode, bnavn, karakter)

Student(bnavn, id)

StudentEmne(bnavn, emnekode, karakter)

Vil du lære mer
relasjonsdatabaseteori?

Ta INF3100 -

Databasesystemer til
våren!