Više normalne forme za relacijske baze podataka

Čičak, Anamaria

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Anamaria Čičak

VIŠE NORMALNE FORME ZA RELACIJSKE BAZE PODATAKA

Diplomski rad

Voditelj rada:
doc. dr. sc. Marko Horvat

Zagreb, studeni 2023.

Ovaj diplomski rad obranjen je dana ____________________ pred ispitnim povjerenstvom u sastavu:

1. ____________________________, predsjednik
2. ____________________________, član
3. ____________________________, član

Povjerenstvo je rad ocijenilo ocjenom ________________.

Potpisi članova povjerenstva:

1. ____________________________
2. ____________________________
3. ____________________________

Veliko hvala majci Snježani, ocu Anti, sestri Iwi te braći Mateju i Ivanu na bezuvjetnoj potpori, ljubavi i strpljenju.
Zahvaljujem mentoru doc. dr. sc. Marku Horvatu na podršci, savjetima i vremenu uloženom u mentoriranje ovog rada.

Sadržaj

Sadržaj **1**

Uvod **2**

1 Osnovni pojmovi i definicije **3**
1.1 Relacijski model baze podataka 3
1.2 Nadključ, kandidat za ključ i primarni ključ 5

2 Niže normalne forme **8**
2.1 Prva, druga i treća normalna forma 9
2.2 Boyce-Coddova normalna forma 15
2.3 Usporedba različitih definicija normalnih formi 15

3 Više normalne forme **18**
3.1 Četvrta normalna forma 18
3.2 Peta normalna forma 20
3.3 Šesta normalna forma 27
3.4 Ostale normalne forme (EKNF, ETNF, DKNF) 28

Bibliografija **41**

Uvod

U današnjem svijetu u kojem je internet postao globalno dostupan, količina podataka koja svakodnevno nastaje postaje sve veća i veća. Svaku poruku koja se pošalje, svaku transakciju koja se obavi, svaki klik u nekoj aplikaciji može biti potrebno pratiti te zapisati u neku bazu podataka. Kako bismo mogli velike količine podataka što lakše i brže administrirati i pregledavati, bitno je bazu podataka oblikovati konzistentno te s minimalnom redundancijom kako ne bismo omogućili pohranjivanje oprečnih podataka ili bespotrebno gomilanje identičnih podataka.

U ovome radu bavit ćemo se procesom **normalizacije** relacijske baze podataka. Normalizacija predstavlja proces u kojem se relacije oblikuje što efikasnije, smanjujući pritom redundanciju te čuvajući konzistentnost pohranjenih podataka. Njome se od svake relacije traži da zadovoljava određene kvalitativne zahtjeve u obliku **normalnih formi**, ovisno o potrebama baze.

Cilj rada bit će izložiti više normalne forme za relacijske baze podataka. Stoga se u prvom poglavlju prisjećamo osnovnih pojmova vezanih uz relacijski model baza podataka. U drugom poglavlju definiramo niže normalne forme do uključivo *Boyce-Coddove normalne forme* te promatramo i uspoređujemo različite definicije nekih od prethodno uvedenih normalnih formi. Na početku trećeg poglavlja obrađujemo više normalne forme — *četvrtu, petu i šestu normalnu formu* — koje predstavljaju završetak tradicionalne normalizacije. U zadnjem dijelu trećeg poglavlja izlažemo ostale normalne forme — *normalnu formu elementarnog ključa, normalnu formu esencijalne n-torke* te *normalnu formu domenskog ključa* — koje će biti zasnovane na prethodno uvedenim normalnim formama te predstavljati njihovo poboljšanje.

Poglavlje 1

Osnovni pojmovi i definicije

Kako bismo mogli objasniti normalne forme u relacijskim bazama podataka, u ovome poglavlju izlažemo osnovne pojmove i definicije vezane uz sam relacijski model baze podataka.

1.1 Relacijski model baze podataka

Relacijski model baze podataka zasnovan je na matematičkom pojmu relacije, točnije baza se promatra kao skup relacija koje su podložne promjenama kroz vrijeme. Pri tome **relaciju** R sa skupom **atributa** $\{A_1, A_2, \ldots, A_n\}$, za $n \in \mathbb{N}_+$, označavamo kao $R(A_1, A_2, \ldots, A_n)$. Svakom atributu A_i, $i \in 1, \ldots, n$, pripada **domena**, u oznaci $dom(A_i)$, koja predstavlja skup vrijednosti koje atribut A_i može poprimiti. Dakle, relaciju definiramo kao proizvoljan podskup Kartezijevog produkta $dom(A_1) \times \ldots \times dom(A_n)$.

Relacija može biti predstavljena **relacijskom tablicom**, pri čemu redak tablice odgovara jednom elementu ili **n-torci** relacije, a stupac atributu relacije. Ako je t n-torka proizvoljne relacije, tada $t(A)$ označava vrijednost atributa A u n-torci t, što nazivamo **A-vrijednost od t**. Ukoliko je X podskup skupa atributa od R, tada **X-vrijednost od t**, u oznaci $t[X]$, predstavlja restrikciju n-torke t na skup X čija je B-vrijednost od t za svaki $B \in X$ jednaka $t(B)$. Nadalje, relaciju $R[X]$ koja je skup svih jedinstvenih n-torki $t[X]$, nazivamo **projekcijom relacije R nad skupom X**.

Stupanj relacije definiramo kao broj atributa relacije, a **kardinalnost relacije** kao broj n-torki relacije.

Pod pojmom **shema relacije** mislimo na ime relacije i skup atributa relacije. Skup svih shema relacija pojedine baze nazivamo **relacijskom shemom baze podataka**.

Napomena 1.1.1. *U ovome radu ćemo pojmove relacija i relacijska tablica smatrati ravnopravnima te ćemo se koristeći jedan implicitno referirati i na drugi. Nadalje, često ćemo* *umjesto relacijska tablica koristiti kraći naziv — tablica.*

Također, pojmove jednočlani skup atributa i atribut ćemo poistovjećivati.

Primjer 1.1.2. *Pokažimo sada na primjeru prethodno definirane pojmove. Promotrimo bazu podataka vezanu uz trgovinu koja je predstavljena sljedećim opisom.*

Blagajnici naplaćuju artikle koje kupuju kupci. Svaku kupovinu naplaćuje točno jedan blagajnik, a obavlja točno jedan kupac. U jednoj kupovini blagajnik može kupcu naplatiti jedan ili više različitih artikala (ne broji se količina pojedinog artikla). Kupac prilikom kupovine u trgovini može kupiti jedan ili više artikala, uz pretpostavku da je kupac kupio barem jedan artikl. Isti artikl može kupiti 0 ili više kupaca. Svaki kupac mora posjedovati točno jednu karticu potrebnu za kupovinu te svaka kartica pripada točno jednom kupcu.

Dakle, relacijsku shemu opisane baze podataka možemo u nekom trenutku u vremenu predstaviti sljedećim relacijama, odnosno relacijskim tablicama `BLAGAJNIK`, `ARTIKL`, `KUPAC`, `KUPOVINA` *i* `RAČUN` *prikazanim u tablicama 1.1–1.5.*

Relacijska shema baze podataka sastoji se od sljedećih shema relacija:

- `BLAGAJNIK(ID_BLAGAJNIKA, IME, PREZIME)`
- `ARTIKL(BARKOD, NAZIV)`
- `KUPAC(ID_KARTICE, OIB, IME, PREZIME)`
- `KUPOVINA(ID_KUPOVINE, ID_BLAGAJNIKA, ID_KARTICE, DATUM_KUPOVINE)`
- `RAČUN(ID_KUPOVINE, BARKOD)`

Atributi relacija navedeni su unutar zagrada, odvojeni zarezom, pa su tako primjerice atributi relacije `BLAGAJNIK`: `ID_BLAGAJNIKA`, `IME` *i* `PREZIME`.

Domene atributa relacija dane su u tablicama 1.6–1.10 (pri čemu NUMBER predstavlja skup cijelih brojeva, CHAR(20) skup riječi maksimalne duljine 20 znakova, a TIMESTAMP(6) vrijeme na razini mikrosekundi).

Za n-torku $t = (77489124347, 23545232152, Marija, Horvat)$ *iz tablice 1.3 i skup atributa* $X = \{\mathtt{ID\_KARTICE}, \mathtt{OIB}, \mathtt{PREZIME}\}$, *X-vrijednost od t jednaka je:*

$$t[X] = (t(\mathtt{ID\_KARTICE}), t(\mathtt{OIB}), t(\mathtt{PREZIME})) = (77489124347, 23545232152, Horvat).$$

Primijetimo da su relacije `KUPAC` *i* `KUPOVINA` *stupnja četiri, relacija* `BLAGAJNIK` *stupnja tri, a relacije* `ARTIKL` *i* `RAČUN` *stupnja dva. Budući da n-torke relacije odgovaraju retcima u tablicama, n-torke relacije* `ARTIKL` *odgovaraju retcima tablice 1.2.*

Sve navedene relacije imaju po tri n-torke, osim relacije `RAČUN` *koja ima četiri. Dakle, navedene relacije su kardinalnosti tri, odnosno četiri.*

1.2 Nadključ, kandidat za ključ i primarni ključ

Uz pojam relacije vežu se pojmovi *nadključ*, *kandidat za ključ* te *primarni ključ* koji imaju značajnu ulogu u definiciji normalnih formi.

Definicija 1.2.1. *Neka je R relacija sa skupom atributa U.* ***Nadključ*** *relacije R je neprazan skup atributa $X \subseteq U$ takav da vrijednosti od X jedinstveno određuju vrijednosti od U.*

Primijetimo da iz prethodne definicije slijedi da različite n-torke relacije ne mogu imati iste vrijednosti nadključa te da je svaki nadskup nadključa (s obzirom na skup atributa relacije) i sam nadključ. Također, primijetimo da svaka relacija ima barem jedan nadključ jer skup svih atributa zadovoljava uvjete iz definicije, stoga ćemo takav nadključ nazivati **trivijalnim**.

Definicija 1.2.2. *Neka je R relacija sa skupom atributa U.* ***Kandidat za ključ*** *relacije R je nadključ K relacije R takav da ne postoji $K' \subsetneq K$ koji je također nadključ relacije R.*

Ako je atribut A relacije R element nekog kandidata za ključ K, tada A nazivamo ***primarnim atributom****, u suprotnom A nazivamo* ***neprimarnim atributom****.*

Prema napomeni 1.1.1, kada jednočlani skup atributa čini bilo nadključ bilo kandidata za ključ, možemo pisati samo naziv tog atributa. Na primjer, za kandidat za ključ $K = \{$`ATR`$\}$ pisat ćemo samo `ATR`, a podrazumijevat ćemo da je K jednočlan skup atributa. Nadalje, kandidata za ključ koji odgovara trivijalnom nadključu nazivat ćemo **trivijalnim**.

Vidjet ćemo u primjeru 1.2.3 da kandidat za ključ relacije ne mora nužno biti jedinstven. Stoga kandidata za ključ koji je najpogodniji za identifikaciju proglašavamo **primarnim ključem**.

Primjer 1.2.3. *Pogledajmo ponovno relacije iz primjera 1.1.2.*

Uočimo da u relaciji `KUPOVINA` trivijalni nadključ odgovara skupu atributa $\{$`ID_KUPOVINE, ID_BLAGAJNIKA, ID_KARTICE, DATUM_KUPOVINE`$\}$, a netrivijalni nadključ primjerice skupu atributa $\{$`ID_KUPOVINE, ID_BLAGAJNIKA`$\}$. Relacija ima samo jedan kandidat za ključ, `ID_KUPOVINE`, koji će ujedno biti i primarni ključ.

Promotrimo li pak relaciju `RAČUN`, primjećujemo da ona ima samo jednog trivijalnog kandidata za ključ, $\{$`ID_KUPOVINE, BARKOD`$\}$. Za primarni ključ odabiremo jedinog kandidata za ključ.

S druge strane, relacija `KUPAC` ima dva kandidata za ključ: `ID_KARTICE` i `OIB`, i to ne sasvim slučajno. Naime, kako bi se zaštitili osobni podaci kupca, a pri tome osigurala jedinstvena identifikacija kupca na blagajni česta je praksa da se svakom kupcu dodjeljuje jedinstven `ID_KARTICE`. Također, budući svaki kupac mora posjedovati točno jednu karticu, možemo pretpostaviti da bi se u slučaju kada bi kupac izgubio karticu njemu izdala nova kartica čiji bi vrijednost zamijenila u relaciji `KUPAC` vrijednost izgubljene kartice.

Budući da je `ID_KARTICE` *za danog kupca podložan promjenama možemo zaključiti da bi kao primarni ključ za ovu relaciju trebao biti izabran* `OIB`*.*

ID_BLAGAJNIKA	IME	PREZIME
1	Ana	Knežević
2	Ivan	Horvat
3	Marko	Marić

Tablica 1.1: Relacija BLAGAJNIK

BARKOD	NAZIV
57328947321	čips
22724099371	kruška
92031172303	kokice

Tablica 1.2: Relacija ARTIKL

ID_KARTICE	OIB	IME	PREZIME
77489124347	23545232152	Marija	Horvat
34624763733	50175878784	Snježana	Posavec
70704854752	28734932029	Iva	Jerković

Tablica 1.3: Relacija KUPAC

ID_KUPOVINE	ID_BLAGAJNIKA	ID_KARTICE	DATUM_KUPOVINE
1	1	77489124347	20/1/2023/ 10:17:35.941871
2	2	34624763733	23/1/2023/ 10:36:24.364234
3	1	34624763733	24/1/2023/ 2:25:33.620710

Tablica 1.4: Relacija KUPOVINA

ID_KUPOVINE	BARKOD
1	57328947321
1	22724099371
2	22724099371
3	92031172303

Tablica 1.5: Relacija RAČUN

NAZIV ATRIBUTA	DOMENA
ID_BLAGAJNIKA	NUMBER
IME	CHAR(20)
PREZIME	CHAR(20)

Tablica 1.6: Domene atributa relacije BLAGAJNIK

NAZIV ATRIBUTA	DOMENA
BARKOD	NUMBER
NAZIV	CHAR(20)

Tablica 1.7: Domene atributa relacije ARTIKL

NAZIV ATRIBUTA	DOMENA
ID_KARTICE	NUMBER
OIB	NUMBER
IME	CHAR(20)
PREZIME	CHAR(20)

Tablica 1.8: Domene atributa relacije KUPAC

NAZIV ATRIBUTA	DOMENA
ID_KUPOVINE	NUMBER
ID_BLAGAJNIKA	NUMBER
ID_KARTICE	NUMBER
DATUM_KUPOVINE	TIMESTAMP(6)

Tablica 1.9: Domene atributa relacije KUPOVINA

NAZIV ATRIBUTA	DOMENA
ID_KUPOVINE	NUMBER
BARKOD	NUMBER

Tablica 1.10: Domene atributa relacije RAČUN

Poglavlje 2

Niže normalne forme

Normalizacija podataka sastavni je dio logičkog oblikovanja baze podataka. Ona predstavlja niz koraka kojim se nastoji ukloniti ili smanjiti redundantnost te olakšati čuvanje konzistentnosti baze. Primjerice, kada bi baza podataka bila realizirana kao jedna relacija, tada bi ona bila sklonija redundanciji podataka, tj. ažurirati ili izbrisati nešto u toj tablici bilo bi sve „skuplje” i sklonije narušavanju konzistentnosti kako bi broj redaka rastao. Stoga, kako bi se smanjila redundantnost podataka, normalizacija će zahtijevati razdvajanje relacijskih tablica u bazi u više manjih poštujući pritom konzistentnost podataka.

U ovome poglavlju prvo se bavimo promatranjem nižih normalnih formi, odnosno *prve, druge, treće* te *Boyce-Coddove normalne forme*. Zatim, u zadnjem dijelu poglavlja promatramo i uspoređujemo različite definicije nekih od prethodno uvedenih normalnih formi.

Promotrimo sljedeći primjer kao motivaciju za uvođenje normalnih formi.

Primjer 2.0.1. *Pretpostavimo da se baza podataka neke web trgovine sastoji samo od jedne tablice, tj. dana web trgovina prati samo podatke o svojim prodanim proizvodima zapisujući ih kao u tablici 2.1.*

Relacijska shema baze podataka predstavljena je samo jednom relacijom `WEB_TRGOVINA(ID, PROIZVOD, IME_PREZIME, BR_MOB, DAT_PROMJENE, STATUS)`.

Na ovom primjeru vidimo da su podaci o narudžbama i kupcima zapisani redundantno. Kako bismo primjerice saznali sve narudžbe pojedinog kupca ili kako bismo saznali tko je prošli mjesec kupio određeni proizvod bit će potrebno pretražiti cijelu tablicu. Nadalje, ako bi bilo potrebno ažurirati podatke o kupcu Ivan Horvat, tada bismo morali ažurirati ukupno tri retka u tablici, što nije efikasno ni na ovako maloj tablici.

Primjećujemo da su time i administriranje i pretraživanje takve baze teži. Stoga, ako bismo ovu tablicu mogli razbiti u više manjih, problem redundantnosti bi se smanjio, a i ažuriranje podataka o nekom korisniku bi se znatno ubrzalo.

Vidjet ćemo kroz ovo poglavlje da upravo normalne forme predstavljaju zahtjeve na relaciju koji joj osiguravaju navedena poželjna svojstava.

ID	PROIZVOD	IME_PREZIME	BR_MOB	DAT_PROMJENE	STATUS
1	sokovnik igrica filter	Ivan Horvat	+385398011541	1.7.2023.	potvrđena
2	cvijet vaza	Marija Kralj	+385389348018	1.7.2023.	u čekanju
3	bicikl boca sat	Ivan Horvat	+385398011541	2.7.2023.	potvrđena
4	parfem tenisice	Katarina Posavec	+385885477093	2.7.2023.	potvrđena
5	olovka	Ivan Horvat	+385398011541	3.7.2023.	u čekanju

Tablica 2.1: Relacija WEB_TRGOVINA

2.1 Prva, druga i treća normalna forma

U ovom potpoglavlju obradit ćemo prvu, drugu i treću normalnu formu koje predstavljaju standard u oblikovanju relacijskih baza podataka te koje su često u praksi i najviše zadovoljene normalne forme.

Napomena 2.1.1. *Očito je da svaka relacija implicira svoju relacijsku shemu, te ćemo stoga pri definiranju normalnih formi reći da relacija zadovoljava normalnu formu, a pri tome ćemo podrazumijevati da njena relacijska shema zadovoljava danu normalnu formu. Jedino ćemo eksplicitno u potpoglavlju 3.4, prilikom definiranja ETNF i DKNF, definirati normalne forme za relacijsku shemu.*

Prva normalna forma

Podzapisom relacije nazivamo atribut koji je sastavljen od drugih atributa. Na primjer, atribut DRŽAVA_MJESTO_ROĐENJA je podzapis koji je sastavljen od atributa DRŽAVA_ROĐENJA i MJESTO_ROĐENJA. Nadalje, vrijednost atributa je **jednostruka** ako odgovara točno jednom elementu domene danog atributa.

Definicija 2.1.2. *Relacija R je u* ***prvoj normalnoj formi (1NF)*** *ako ne sadrži podzapise te su sve vrijednosti svih atributa jednostruke.*

Proučavajući relacijski model baze podataka možemo zaključiti da je svojstvo prve normalne forme automatski ugrađeno u sami model. Ipak, razlog zbog kojeg se definira prva normalna forma proizlazi iz potrebe za konverzijom podataka koji su pohranjeni u nekom drugom obliku. Primjerice, ako želimo podatke pohranjene u Excel tablici pohraniti u relacijsku bazu podataka, tada moramo na osnovu te tablice oblikovati relacije koje će zadovoljavati prvu normalnu formu.

Transformacija u 1NF

Postupak transformacije u 1NF prati sljedeće korake. Za svaki atribut relacije:

1. Ako atribut predstavlja podzapis te imamo potrebu pristupati atributima od kojih je sastavljen, tada atribut zamijenimo atributima od kojih je sastavljen. U suprotnome, atribut (podzapis) smatramo jednim nedjeljivim atributom.

2. Za n-torke u kojima vrijednost nekog atributa nije jednostruka dodajemo onoliko novih n-torki koliko je vrijednosti bilo upisano za dani atribut u promatranoj n-torci, pri čemu u novim n-torkama vrijednosti ostalih atributa relacije ostaju identične.

Primjer 2.1.3. *U ovom primjeru izlažemo postupak prevođenja tablice iz uvodnog primjera 2.0.1, gdje se možda radi o Excel tablici, u prvu normalnu formu.*

`ID`	`PROIZVOD`	`IME`	`PREZIME`	`BR_MOB`	`DAT_PROMJENE`	`STATUS`
1	sokovnik	Ivan	Horvat	+385398011541	1.7.2023.	potvrđena
1	igrica	Ivan	Horvat	+385398011541	1.7.2023.	potvrđena
1	filter	Ivan	Horvat	+385398011541	1.7.2023.	potvrđena
2	cvijet	Marija	Kralj	+385389348018	1.7.2023.	u čekanju
2	vaza	Marija	Kralj	+385389348018	1.7.2023.	u čekanju
3	bicikl	Ivan	Horvat	+385398011541	2.7.2023.	potvrđena
3	boca	Ivan	Horvat	+385398011541	2.7.2023.	potvrđena
3	sat	Ivan	Horvat	+385398011541	2.7.2023.	potvrđena
4	parfem	Katarina	Posavec	+385885477093	2.7.2023.	potvrđena
4	tenisice	Katarina	Posavec	+385885477093	2.7.2023.	potvrđena
5	olovka	Ivan	Horvat	+385398011541	3.7.2023.	u čekanju

Tablica 2.2: Relacija `WEB_TRGOVINA` u 1NF

Atribut (podzapis) `IME_PREZIME` *smo rastavili na dva atributa* `IME` *i* `PREZIME` *jer je u danoj relaciji bitno imati mogućnost odvojenog administriranja i imena i prezimena kupca (može se recimo dogoditi da kupac promijeni prezime, pa će biti potrebno ažurirati samo njegovo prezime, dok bismo kupca u komunikaciji možda htjeli oslovljavati samo njegovim imenom).*

Atribut (podzapis) `DAT_PROMJENE` *nismo rastavili na atribute* `DAN_PROMJENE`, `MJESEC_PROMJENE` *i* `GODINA_PROMJENE` *jer smatramo da nemamo potrebu pristupati njima* *pojedinačno. Dakle, atribut* `DAT_PROMJENE` *smatramo jednim nedjeljivim atributom za relaciju predstavljenu tablicom 2.2.*

Također, sve n-torke koje su sadržavale više od jedne vrijednosti za atribut `PROIZVOD` *rastavili smo na onoliko n-torki koliko je vrijednosti bilo upisano u n-torci za atribut* `PROIZVOD`*, s time da su vrijednosti ostalih atributa ostale identične u odnosu na početnu n-torku. Tako smo primjerice n-torku čija je vrijednost atributa* `ID` *jednaka 3 te koja za atribut* `PROIZVOD` *ima vrijednosti bicikl, boca i sat rastavili na sljedeće tri n-torke:*

- *(3, bicikl, Ivan, Horvat, +385398011541, 2.7.2023., potvrđena)*
- *(3, boca, Ivan, Horvat, +385398011541, 2.7.2023., potvrđena)*
- *(3, sat, Ivan, Horvat, +385398011541, 2.7.2023., potvrđena)*

Druga normalna forma

Kako bismo mogli uvesti i objasniti koncept druge normalne forme te viših normalnih formi, najprije je potrebno definirati pojmove ovisnosti među atributima. Definirat ćemo pojmove *funkcionalne ovisnosti*, *potpune funkcionalne ovisnosti* te *parcijalne funkcionalne ovisnosti* u relacijskim bazama podataka, a zatim i samu drugu normalnu formu.

Definicija 2.1.4. *Za (neprazne) skupove atributa A i B kažemo:*

- *B je **funkcionalno ovisan** o A ako vrijednosti od A jedinstveno određuje vrijednosti od B; pišemo $A \to B$.*
- *B je **potpuno funkcionalno ovisan** o A ako $A \to B$, i ne postoji $A' \subsetneq A$ takav da $A' \to B$.*
- *B je **parcijalno funkcionalno ovisan** o A ako $A \to B$, ali postoji $A' \subsetneq A$ takav da $A' \to B$.*

*Funkcionalnu ovisnost $A \to B$ nazivamo **trivijalnom** ako je $B \subseteq A$. Inače, kažemo da je **netrivijalna**.*

Napomena 2.1.5. *Slijedeći literaturu, za funkcionalnu ovisnost ćemo ponekad kraće pisati **FD** (engl. functional dependency).*

Činjenicu da skup atributa B nije funkcionalno ovisan o skupu atributa A pišemo kao $A \nrightarrow B$.

Funkcionalna ovisnost posjeduje sljedeća svojstva (za više detalja vidi [9]). Neka su X, Y, Z i W neprazni skupovi atributa.

- (F1, refleksivnost) ako je $Y \subseteq X$, tada $X \to Y$
- (F2) ako je $Z \subseteq W$ i $X \to Y$, tada $X \cup W \to Y \cup Z$
- (F3, tranzitivnost) ako $X \to Y$ i $Y \to Z$, tada $X \to Z$

Razna svojstva FD mogu biti izvedena iz prethodno navedenih, primjerice:

- (F4, dekompozicija) ako $X \to Y$, tada $X \to A$ za svaki $A \in Y$.

Primjer 2.1.6. *U primjeru 1.1.2 u relaciji* `KUPAC` *uočavamo sljedeće netrivijalne funkcionalne ovisnosti:*

- $\mathtt{ID\_KARTICE} \to \mathtt{OIB}$, $\mathtt{ID\_KARTICE} \to \mathtt{IME}$, $\mathtt{ID\_KARTICE} \to \mathtt{PREZIME}$
- $\mathtt{OIB} \to \mathtt{ID\_KARTICE}$, $\mathtt{OIB} \to \mathtt{IME}$, $\mathtt{OIB} \to \mathtt{PREZIME}$

koje su i potpune funkcionalne ovisnosti.

Druga normalna forma predstavlja jače zahtjeve u odnosu na prvu normalnu formu. Naime, kako bi relacija zadovoljava drugu normalnu formu, ona po definiciji mora zadovoljavati i prvu. Općenito, svaka kasnija normalna forma mora zadovoljavati sve normalne forme koje joj prethode (po definiciji ili zbog svojih svojstava).

Definicija 2.1.7. *Relacija R je u* ***drugoj normalnoj formi (2NF)*** *ako je u 1NF i ako je svaki neprimarni atribut od R potpuno funkcionalno ovisan o svakom kandidatu za ključ relacije R.*

U sljedećem primjeru predstavljamo relaciju koja jest u prvoj normalnoj formi, no nije u drugoj, te obrazlažemo potrebu za uvođenjem druge normalne forme.

Primjer 2.1.8. *Promotrimo relaciju* `PROIZVOD_IZVOZ(ID_PROIZVODA, ID_IZVOZNIKA, NAZIV_IZVOZNIKA, ZEMLJA_IZVOZA)` *predstavljenu tablicom 2.3.*

ID_PROIZVODA	ID_IZVOZNIKA	NAZIV_IZVOZNIKA	ZEMLJA_IZVOZA
1	1	Spain d.o.o	Španjolska
2	1	Spain d.o.o	Španjolska
1	2	Italy d.o.o	Italija
2	2	Italy d.o.o	Italija
3	2	Italy d.o.o	Italija
3	3	Madrid d.o.o	Španjolska

Tablica 2.3: Relacija `PROIZVOD_IZVOZ` u 1NF, ali ne i u 2NF

Pretpostavimo da jedan proizvod može izvoziti više kompanija te da jedna kompanija izvozi u točno jednu zemlju. Uočimo kandidate za ključ {`ID_PROIZVODA, ID_IZVOZNIKA`} *i* {`ID_PROIZVODA, NAZIV_IZVOZNIKA`}. *Primijetimo da je relacija u 1NF i da neprimarni atribut* `ZEMLJA_IZVOZA` *parcijalno ovisi o kandidatima za ključ. Zaključujemo da relacija* `PROIZVOD_IZVOZ` *nije u 2NF.*

Prije nego što prevedemo danu relaciju u 2NF, zapitajmo se sljedeće. Što bi se dogodilo kada bi u jednome trenutku izvoznik Madrid d.o.o. prestao izvoziti proizvod 3 zbog poslovne odluke (primjerice, transport vrlo teškog proizvoda 3 je preskup)? Kompanija čije proizvode izvozi bi zasigurno htjela zadržati podatke o Madrid d.o.o. zbog moguće ponovne suradnje u budućnosti, no podatak kamo ta firma izvozi bi se nepovratno izgubio brisanjem zadnjeg retka iz tablice `PROIZVOD_IZVOZ`.

Transformacija iz 1NF u 2NF

Kako bismo relaciju iz 1NF preveli u 2NF, ponavljamo sljedeće korake.
Za svaki kandidat za ključ K relacije R, za svaku netrivijalnu funkcionalnu ovisnost $X \to A$ relacije R koja ne ispunjava uvjete iz definicije 2.1.7 (tj. za koju vrijedi da je $X \subsetneq K$ i A neprimarni atribut):

1. Ako ne postoji relacija R' s kandidatom za ključ X, tada stvaramo takvu relaciju.
2. U relaciju R' dodajemo neprimarni atribut A.
3. Iz relacije R uklonimo sve neprimarne atribut A.

Primjer 2.1.9. *Kao što smo i ustanovili u primjeru 2.1.8, relacija* `PROIZVOD_IZVOZ` *nije u 2NF jer jedini neprimarni atribut* `ZEMLJA_IZVOZA` *nije potpuno funkcionalno ovisan o svakom (štoviše, niti o jednom) kandidatu za ključ. Naime, vrijedi* `ID_IZVOZNIKA` $\to$ `ZEMLJA_IZVOZA` *te* `NAZIV_IZVOZNIKA` $\to$ `ZEMLJA_IZVOZA`.

Prateći postupak prevođenja iz 1NF u 2NF, relacija `PROIZVOD_IZVOZ` *će biti rastavljena na relacije:*

- `PROIZVOD_IZVOZI(ID_PROIZVODA, ID_IZVOZNIKA, NAZIV_IZVOZNIKA)`
- `IZVOZNIK(ID_IZVOZNIKA, ZEMLJA_IZVOZA)`

ili na relacije:

- `PROIZVOD_IZVOZI(ID_PROIZVODA, ID_IZVOZNIKA, NAZIV_IZVOZNIKA)`
- `IZVOZNIK(NAZIV_IZVOZNIKA, ZEMLJA_IZVOZA)`

ovisno o tome je li postupak prevođenja iz 1NF u 2NF proveden za
K = {`ID_PROIZVODA, ID_IZVOZNIKA`} *i* `ID_IZVOZNIKA` $\to$ `ZEMLJA_IZVOZA`, *ili za*
K = {`ID_PROIZVODA, NAZIV_IZVOZNIKA`} *i* `NAZIV_IZVOZNIKA` $\to$ `ZEMLJA_IZVOZA`.

Treća normalna forma

Definicija 2.1.10. *Relacija R je u **trećoj normalnoj formi (3NF)** ako je u 2NF i za svaku netrivijalnu funkcionalnu ovisnost $X \to A$ vrijedi:*

- *X je nadključ ili*
- *svi elementi skupa $A \setminus X$ su primarni atributi*

Transformacija iz 2NF u 3NF

Za svaku netrivijalnu funkcionalnu ovisnost $X \to A$ koja ne ispunjava uvjete iz definicije 2.1.10:

1. Ako ne postoji relacija R' s kandidatom za ključ X, tada stvaramo takvu relaciju.
2. U relaciju R' dodajemo atribute iz $A \setminus X$.
3. Iz relacije R uklonimo atribute iz $A \setminus X$.

Primjer 2.1.11. *Kao primjer prevođenja u 3NF preoblikujmo sljedeću relaciju:*
`ZAPOSLENIK(OIB, IME, PREZIME, ID_ODJELA, OIB_VODITELJA)`.
Relacija `ZAPOSLENIK` *izražava činjenicu da svaki zaposlenik pripada točno jednom odjelu te da svaki odjel, odnosno zaposlenik ima točno jednog voditelja. Neka relacija ZAPOSLENIK zadovoljava sljedeće netrivijalne funkcionalne ovisnosti:*

- `OIB` $\to$ `IME, PREZIME, ID_ODJELA, OIB_VODITELJA`
- `ID_ODJELA` $\to$ `OIB_VODITELJA`

Primijetimo da je jedini kandidat za ključ ove relacije `OIB`*. Stoga su svi ostali atributi neprimarni. Uočimo FD* `ID_ODJELA` $\to$ `OIB_VODITELJA` *zbog koje relacija nije u 3NF te primijenimo postupak prevođenja iz 2NF u 3NF. Dobivamo relacije:*

- `ZAPOSLENIK(OIB, IME, PREZIME, ID_ODJELA)`
- `ODJEL(ID_ODJELA, OIB_VODITELJA)`

koje zadovoljavaju 3NF.

2.2 Boyce-Coddova normalna forma

Za razliku od 3NF, Boyce-Coddova normalna forma dozvoljava da se na lijevoj strani netrivijalnih funkcionalnih ovisnosti nalazi samo nadključ relacije.

Definicija 2.2.1. *Relacija R je u **Boyce-Coddovoj normalnoj formi (BCNF)** ako za svaku netrivijalnu funkcionalnu ovisnost $X \to A$ vrijedi da je X nadključ relacije R.*

Transformacija iz 3NF u BCNF

Primijetimo da će transformacija relacije R iz 3NF u BCNF slijediti slične korake kao što bi transformacija relacije R iz 2NF u 3NF. Za svaku netrivijalnu funkcionalnu ovisnost $X \to A$ koja ne ispunjava uvjete iz definicije 2.2.1:

1. Ako ne postoji relacija R' s nadključem X, tada stvaramo takvu relaciju.
2. U relaciju R' dodajemo atribute iz $A \setminus X$.
3. Iz relacije R uklonimo atribute iz $A \setminus X$.

Primjer 2.2.2. *Neka je dana relacija*

`STUDENT_MENTOR(JMBAG, OIB_STUDENTA, OIB_MENTORA)`

koja izražava činjenicu da jedan student može imati jednog ili više mentora za razne radove koje piše, s pretpostavkom da se mentor pridružuje studentu ako je ikada mentorirao neki njegov rad. Kandidati za ključ su skupovi atributa {`JMBAG, OIB_MENTORA`} i {`OIB_STUDENTA, OIB_MENTORA`}. Dakle, svi atributi relacije su primarni, pa relacija zadovoljava i 2NF i 3NF. U relaciji su, između ostalih, prisutne funkcionalne ovisnosti `JMBAG` → `OIB_STUDENTA` te `OIB_STUDENTA` → `JMBAG` koje narušavaju BCNF jer niti `OIB_STUDENTA` niti `JMBAG` nisu nadključevi relacije.

Sada transformacijom u BCNF dobivamo sljedeće relacije:

- `STUDENT_MENTOR(JMBAG, OIB_MENTORA)` *ili* `STUDENT_MENTOR(OIB_STUDENTA, OIB_MENTORA)`
- `STUDENT(JMBAG, OIB_STUDENTA)`

Primijetimo da ako relacija zadovoljava BCNF, tada ona zadovoljava i 3NF.

2.3 Usporedba različitih definicija normalnih formi

Postoje mnoge različite definicije normalnih formi, no eventualne razlike se izgube prelaskom na kasnije normalne forme. U ovom poglavlju dat ćemo usporedbu definicije 2NF u ovome radu koja je preuzeta iz [2] te one iz [7], a zatim ćemo dati i jednu ekvivalentnu definiciju 3NF po uzoru na [2].

Usporedba definicija druge normalne forme

Promotrimo najprije definiciju 2NF danu u [7] te ju usporedimo s definicijom 2.1.7.

Definicija 2.3.1. *Kažemo da je relacija R u* ***drugoj normalnoj formi (2NF)*** *ako je u 1NF i ako je svaki atribut koji nije element primarnog ključa od R je potpuno funkcionalno ovisan o primarnom ključu.*

Prisjetimo se sljedeće relacije iz primjera 2.1.9:

`PROIZVOD_IZVOZI(ID_PROIZVODA, ID_IZVOZNIKA, NAZIV_IZVOZNIKA).`

Kao što je već rečeno u samom primjeru, imamo dva kandidata za ključ: {`ID_PROIZVODA, ID_IZVOZNIKA`} i {`ID_PROIZVODA, NAZIV_IZVOZNIKA`}. Proglasimo {`ID_PROIZVODA, ID_IZVOZNIKA`} primarnim ključem. Tada, atribut `NAZIV_IZVOZNIKA` jedini nije element primarnog ključa te vrijedi da je `NAZIV_IZVOZNIKA` parcijalno ovisan o primarnom ključu. Po definiciji 2.3.1 relacija ne zadovoljava uvjete 2NF, dok se po definiciji 2.1.7 nalazi u 2NF.

Opisujemo korake transformacije relacije R s primarnim ključem P u 2NF prema definiciji 2.3.1. Ponavljamo sljedeće korake za svaki neprazan skup atributa $P' \subsetneq P$:

1. Ako postoji barem jedan atribut koji nije element primarnog ključa P, a koji potpuno funkcionalno ovisi o P', definiramo novu relaciju R'.

2. Relacija R' sadrži sve atribute iz P' te sve atribute koji nisu elementi primarnog ključa P iz R i potpuno funkcionalno ovise o P'. Primarni ključ je P'.

3. Iz relacije R uklonimo sve atribute relacije R' koji nisu elementi primarnog ključa P'.

Slijedeći navedenu transformaciju, iz početne relacije `PROIZVOD_IZVOZI` dobivamo relacije `PROIZVOD_IZVOZI(ID_PROIZVODA, ID_IZVOZNIKA)` i `IZVOZNIK(ID_IZVOZNIKA, NAZIV_IZVOZNIKA)` koje zadovoljavaju 2NF prema objema uspoređivanim definicijama.

Primijetimo da su definicije 2.1.7 i 2.3.1 istovjetne ako relacija ima jedan kandidat za ključ ili su svi kandidati za ključ relacije jednočlani.

Usporedba definicija treće normalne forme

Prije nego što predstavimo još jednu definiciju 3NF, bit će potrebno definirati *tranzitivnu ovisnost*.

Definicija 2.3.2. *Neka je R relacija sa skupom atributa U te neka su X i Z neprazni podskupovi od U. Kažemo da je Z* ***tranzitivno ovisan*** *o X (s obzirom na R) ako postoji neprazan podskup Y od U takav da vrijedi: $X \to Y$, $Y \nrightarrow X$, $Y \to Z$ te $Y \cap Z = \emptyset$.*

Primijetimo da iz $Y \cap Z = \emptyset$, slijedi da je $Y \to Z$ netrivijalna funkcionalna ovisnost.

Definicija 2.3.3. *Relacija R je u* ***trećoj normalnoj formi (3NF)*** *ako je u 2NF i ako za svaki neprimarni atribut A i kandidat za ključ X relacije R vrijedi da A nije tranzitivno ovisan o X.*

Lema 2.3.4. *Relacija zadovoljava definiciju 2.1.10 ako i samo ako zadovoljava definiciju 2.3.3.*

Dokaz. Neka je R relacija sa skupom atributa U.

Neka R zadovoljava definiciju 2.1.10 i pretpostavimo suprotno, tj. neka je A neprimarni atribut relacije R koji je tranzitivno ovisan o kandidatu za ključ X. Tada postoji neprazan podskup Y od U takav da vrijedi $Y \nrightarrow X$ te $Y \to A$. Također, slijedi da je $Y \to A$ netrivijalna FD. No, R zadovoljava definiciju 2.1.10 i A je neprimarni atribut, pa je Y nadključ relacije R. Slijedi $Y \to X$. Kontradikcija.

S druge strane, neka R zadovoljava definiciju 2.3.3 i neka je $X \to A$ netrivijalna FD relacije R. Pretpostavimo suprotno, tj. neka X nije nadključ od R i neka je $B \in A \setminus X$ neprimarni atribut. Tada vrijedi i $X \to B$. Neka je Y kandidat za ključ. Tada vrijedi $Y \to X$, no X nije nadključ pa $X \nrightarrow Y$. Dakle, neprimarni atribut B je tranzitivno ovisan o kandidatu za ključ Y. Kontradikcija. □

Na kraju uočimo da iz definicije 2.3.3 slijedi da svaka relacija u 3NF posjeduje sljedeće svojstvo: svaki neprimarni atribut od R je potpuno funkcionalno ovisan o svakom kandidatu za ključ (slijedi iz 2NF) i nije tranzitivno ovisan niti o jednom kandidatu za ključ.

Dakle, pomoću 3NF iz relacije se uklanja mogućnost tranzitivne ovisnosti neprimarnog atributa o nekom kandidatu za ključ, odnosno svaki atribut koji tranzitivno ovisi o kandidatu za ključ u relaciji u 3NF mora biti primarni atribut.

Poglavlje 3

Više normalne forme

Normalne forme u prethodnom poglavlju bile su definirane koristeći funkcionalne ovisnosti među skupovima atributa. Kako bismo mogli definirati više normalne forme bit će potrebno uvesti nove koncepte ovisnosti među atributima.

U ovom poglavlju proučit ćemo višeznačnu i spojnu ovisnost među atributima te ćemo pomoću njih definirati četvrtu, petu i šestu normalnu formu. Valja napomenuti da ako se u relaciji ne nailazi na netrivijalnu višeznačnu i spojnu ovisnost, te ako ona zadovoljava BCNF, automatski zadovoljava i šestu normalnu formu.

Prvo ćemo definirati višeznačnu ovisnost te četvrtu normalnu formu, a potom i spojnu ovisnost i petu normalnu formu. Na kraju poglavlja predstavit ćemo i ostale normalne forme: EKNF, ETNF i DKNF koje se redom nalaze između treće i Boyce-Coddove normalne forme, četvrte i pete normalne forme te pete i šeste normalne forme. Tako će primjerice relacija koja zadovoljava EKNF zadovoljavati i 3NF, no ne i BCNF te će imati podjednako „dobra" svojstva kao i BCNF.

3.1 Četvrta normalna forma

Četvrta normalna forma predstavlja korak dalje u normalizaciji podataka. Promatrat ćemo relacije u kojima se nailazi na višeznačnu ovisnost.

Definicija 3.1.1. *Grupirajmo sve atribute relacije R u disjunktne neprazne skupove A, B i C. Kažemo da postoji **višeznačna ovisnost od A do B**, u oznaci $A \twoheadrightarrow B$, ako vrijedi: ako postoje n-torke t_1, t_2 relacije R takve da je $t_1[A] = t_2[A]$, tada postoji i n-torka t u R takva da $t = (t_1[A], t_1[B], t_2[C])$.*

Ekvivalentno je:

- $A \twoheadrightarrow B$

- $A \twoheadrightarrow C$

Napomena 3.1.2. *Slijedeći literaturu, za višeznačnu ovisnost često ćemo kraće pisati **MVD** (engl. multivalued dependency).*

Ako je skup atributa u prethodnoj definiciji jednočlan, tada u oznaci višeznačne ovisnosti pišemo samo njegovo ime. Na primjer, umjesto $A \twoheadrightarrow \{\mathtt{ATR}\}$ pisat ćemo $A \twoheadrightarrow \mathtt{ATR}$.

Definicija 3.1.3. *Neka je R relacija sa skupom atributa U i $X \twoheadrightarrow Y$ višeznačna ovisnost relacije R. Kažemo da je MVD $X \twoheadrightarrow Y$ **trivijalna** ako je $Y \subseteq X$ ili $X \cup Y = U$.*

Ispitajmo u sljedećem primjeru za dane relacije u kojima se pojavljuju višeznačne ovisnosti, a u kojima ne.

Primjer 3.1.4. *Neka su tablicama 3.1–3.4 dane relacije R1, R2, R3 i R4 sa skupom atributa {X, Y, Z} pri čemu domene atributa X, Y i Z odgovaraju skupu $\mathbb{N}_+$.*

X	Y	Z
1	1	1
1	1	2
1	2	1
1	2	2
2	1	1
2	1	2
3	3	3

Tablica 3.1: Relacija R1

X	Y	Z
1	1	1
1	1	2
2	2	1
2	2	2
3	1	2
3	2	2
4	4	4

Tablica 3.2: Relacija R2

X	Y	Z
1	1	1
1	1	2
1	2	1
3	3	3

Tablica 3.3: Relacija R3

X	Y	Z
1	1	1
1	2	1
2	1	1
2	2	2

Tablica 3.4: Relacija R4

Uočimo da u relacijama R1 i R2 postoji višeznačna ovisnost $\mathtt{X} \twoheadrightarrow \mathtt{Y}$ (ekvivalentno $\mathtt{X} \twoheadrightarrow \mathtt{Z}$).

S druge strane, u relacijama R3 i R4 ne postoje višeznačne ovisnosti. Razlog zbog kojeg R3 ne zadovoljava MVD $\mathtt{X} \twoheadrightarrow \mathtt{Y}$ (ekvivalentno ni MVD $\mathtt{X} \twoheadrightarrow \mathtt{Z}$) je taj što relacija sadrži n-torke (1,1,2) i (1,2,1), no ne i n-torku (1,2,2). Slično, možemo zaključiti da u R4 nedostaju n-torke (2,1,2) i (2,2,1) kako bi u njoj postojala spomenuta višeznačna ovisnost.

Nakon što smo uveli i dali primjer višeznačne ovisnosti kao motivaciju za uvođenjem normalne forme iznad BCNF, pogledajmo na primjeru kako izgleda relacija u BCNF u kojoj postoji višeznačna ovisnost.

Primjer 3.1.5. *Promotrimo relaciju* `KUPAC(ID_KUPCA, KONTAKT, BR_NARUDŽBE)` *za koju vrijedi da jedan kupac može imati više kontakt brojeva (kontakata) na koje ga se može kontaktirati te više različitih kupaca može upisati isti kontakt broj. Također, jedan kupac može napraviti više narudžbi.*

Relacija se nalazi u BCNF jer ne postoje netrivijalne funkcionalne ovisnosti među atributima te je jedini kandidat za ključ skup svih atributa relacije. Također, uočimo da u relaciji postoje sljedeće netrivijalne višeznačne ovisnosti:

- `ID_KUPCA` $\twoheadrightarrow$ `KONTAKT`
- `ID_KUPCA` $\twoheadrightarrow$ `BR_NARUDŽBE`

Skupovi atributa {`KONTAKT`} *i* {`BR_NARUDŽBE`} *ne ovise jedan o drugome, no ako kupac realizira novu narudžbu, tada se u tablicu mora dodati onoliko novih redaka koliko on ima kontakata. Analogno, ako kupac obriše ili promijeni neki kontakt, tada se mora obrisati ili ažurirati onoliko redaka koliko je dani kupac ostvario narudžbi.*

Definicija 3.1.6. *Relacija R je u* ***četvrtoj normalnoj formi (4NF)*** *ako je u BCNF i za svaku višeznačnu ovisnost $X \twoheadrightarrow Y$ u R vrijedi jedno od sljedećeg:*

- *višeznačna ovisnost je trivijalna ili*
- *X je nadključ relacije R.*

Razdvajanjem relacije `KUPAC` na relacije `KUPAC(ID_KUPCA, KONTAKT)` i `KUPAC_NARUDŽBA(ID_KUPCA, BR_NARUDŽBE)` dobivamo dvije relacije koje zadovoljavaju 4NF te u kojima ne nailazimo na ranije spomenute probleme.

3.2 Peta normalna forma

Peta normalna forma poznata je i pod engleskim nazivom *projection-join normal form*. Stoga se na početku ovoga potpoglavlja bavimo spojnom ovisnošću te dekompozicijom relacije, a zatim pomoću tih pojmova dajemo definiciju pete normalne forme.

Spojna ovisnost i dekompozicija

Definicija 3.2.1. *Neka je zadana relacija R_1 sa skupom atributa A i relacija R_2 sa skupom atributa B te neka vrijedi $A \cap B = C$.* ***Prirodni spoj*** *(engl. natural join)* ***relacija*** R_1 ***i*** R_2 *je relacija R sa skupom atributa $A \cup B$ takva da $r = (r_1, r_2) \in R$ ako i samo ako vrijedi: $r_1 \in R_1$, $r_2 \in R_2$ te $r_1(C) = r_2(C)$.*

Napomena 3.2.2. *Važno je napomenuti da u shemi relacija koja je nastala primjenom prirodnog spoja se atributi iz skupa C pojavljuju točno jednom.*

Primijetimo i da ako za skup C iz prethodne definicije vrijedi $C = \emptyset$, tada provedbom prirodnog spoja dobivamo Kartezijev produkt relacija R_1 i R_2.

Primjer 3.2.3. *Neka je tablicom 3.5 dana relacija* `ZAPOSLENIK` *sa skupom atributa A = {`ID_ZAP, KORISNIČKO_IME, ID_ODJELA`} te tablicom 3.6 relacija* `ODJEL` *sa skupom atributa B = {`ID_ODJELA, NAZIV_ODJELA`}.*

ID_ZAP	KORISNIČKO_IME	ID_ODJELA
1	zap_jedan	1
2	zap_dva	1
3	zap_tri	2

Tablica 3.5: Relacija `ZAPOSLENIK`

ID_ODJELA	NAZIV_ODJELA
1	informatika
2	analitika
3	marketing

Tablica 3.6: Relacija `ODJEL`

Provedimo prvo prirodni spoj relacija `ZAPOSLENIK` *i* `ODJEL` *pri čemu primijetimo da je skup C iz definicije 3.2.1 odgovara skupu $A \cap B$ = {`ID_ODJELA`}. Dobivamo relaciju danu tablicom 3.7.*

ID_ZAP	KORISNIČKO_IME	ID_ODJELA	NAZIV_ODJELA
1	zap_jedan	1	informatika
2	zap_dva	1	informatika
3	zap_tri	2	analitika

Tablica 3.7: Relacija dobivena prirodnim spojem relacija `ZAPOSLENIK` i `ODJEL`

Razmotrimo sada projekciju relacije `ZAPOSLENIK` *nad skupom atributa X = {`ID_ZAP, KORISNIČKO_IME`} te provedimo njezin prirodni spoj sa relacijom* `ODJEL`.

Jer relacije `ZAPOSLENIK[X]` *i* `ODJEL` *nemaju zajedničke atributa, odnosno skup C iz definicije 3.2.1 je jednak* $\emptyset$*, provedbom prirodnog spoja danih relacija dobivamo njihov Kartezijev produkt, tj. relaciju iz tablice 3.8.*

ID_ZAP	KORISNIČKO_IME	ID_ODJELA	NAZIV_ODJELA
1	zap_jedan	1	informatika
1	zap_jedan	2	analitika
1	zap_jedan	3	marketing
2	zap_dva	1	informatika
2	zap_dva	2	analitika
2	zap_dva	3	marketing
3	zap_tri	1	informatika
3	zap_tri	2	analitika
3	zap_tri	3	marketing

Tablica 3.8: Relacija nastala primjenom prirodnog spoja relacija `ZAPOSLENIK[{ID_ZAP, KORISNIČKO_IME}]` i `ODJEL`

Definicija 3.2.4. *Neka je* $n \in \mathbb{N}, n \geq 2$*, neka su* $C_1, \ldots, C_n$ *skupovi atributa te neka je* $A = C_1 \cup \ldots \cup C_n$*. Kažemo da* ***relacija R sa skupom atributa A zadovoljava spojnu ovisnost*** $\bowtie\{C_1, \ldots, C_n\}$ *ako relacija R odgovara prirodnom spoju svojih projekcija* $R[C_1], \ldots, R[C_n]$*.*

Relacija R sa skupom atributa A ***trivijalno*** *zadovoljava spojnu ovisnost* $\bowtie\{C_1, \ldots, C_n\}$ *ako je* $A \in \{C_1, \ldots, C_n\}$*.*

Napomena 3.2.5. *Slijedeći literaturu, za spojnu ovisnost često ćemo kraće pisati* ***JD*** *(engl. join dependency).*

U sljedećem primjeru promotrimo spojnu ovisnost relacije.

Primjer 3.2.6. *Neka je relacija* `TEČAJ(NAZIV, LOKACIJA, OIB_SUDIONIKA)` *predstavljena tablicom 3.9.*

NAZIV	LOKACIJA	OIB_SUDIONIKA
Napredni C++	Zagreb	1
Napredni C++	Zagreb	2
Osnove SQL-a	Zagreb	3
Osnove SQL-a	Split	4

Tablica 3.9: Relacija `TEČAJ`

Promotrimo sljedeće spojne ovisnosti:

1. $\bowtie\{\{NAZIV\}, \{LOKACIJA\}, \{OIB\_SUDIONIKA\}\}$

2. $\bowtie\{\{NAZIV, LOKACIJA\}, \{LOKACIJA, OIB\_SUDIONIKA\}\}$

3. $\bowtie\{\{NAZIV, LOKACIJA, OIB\_SUDIONIKA\}, \{OIB\_SUDIONIKA\}\}$

4. $\bowtie\{\{NAZIV, LOKACIJA\}, \{NAZIV, OIB\_SUDIONIKA\}, \{LOKACIJA, OIB\_SUDIONIKA\}\}$

te za svaku od njih odredimo zadovoljava li ih relacija `TEČAJ`*. Dakle, potrebno je provjeriti odgovara li relacija* `TEČAJ` *prirodnom spoju odgovarajućih projekcija iz definicije JD.*

NAZIV	LOKACIJA	OIB_SUDIONIKA
Napredni C++	Zagreb	1
Napredni C++	Zagreb	2
Napredni C++	Zagreb	3
Osnove SQL-a	Zagreb	1
Osnove SQL-a	Zagreb	2
Osnove SQL-a	Zagreb	3
Osnove SQL-a	Split	4

Tablica 3.10: Relacija nastala primjenom prirodnog spoja nad projekcijama `TEČAJ[{NAZIV, LOKACIJA}]` i `TEČAJ[{LOKACIJA, OIB_SUDIONIKA}]`

NAZIV	LOKACIJA	OIB_SUDIONIKA
Napredni C++	Zagreb	1
Napredni C++	Zagreb	2
Osnove SQL-a	Zagreb	3
Osnove SQL-a	Zagreb	4
Osnove SQL-a	Split	3
Osnove SQL-a	Split	4

Tablica 3.11: Relacija nastala primjenom prirodnog spoja nad projekcijama `TEČAJ[{NAZIV, LOKACIJA}]` i `TEČAJ[{NAZIV, OIB_SUDIONIKA}]`

NAZIV	LOKACIJA	OIB_SUDIONIKA
Napredni C++	Zagreb	1
Napredni C++	Zagreb	2
Osnove SQL-a	Zagreb	3
Osnove SQL-a	Split	4

Tablica 3.12: Relacija nastala primjenom prirodnog spoja nad relacijom iz tablice 3.11 i `TEČAJ[{LOKACIJA, OIB_SUDIONIKA}]`

Uočimo najprije da će prirodni spoj projekcija `TEČAJ[{NAZIV}]`, `TEČAJ[{LOKACIJA}]` *te* `TEČAJ[{OIB_SUDIONIKA}]` *biti jednak njihovom Kartezijevom i time različit od relacije* `TEČAJ`, *stoga relacija* `TEČAJ` *ne zadovoljava JD pod 1.*

Promotrimo sada JD pod 2 te provedimo prirodni spoj projekcija `TEČAJ[{NAZIV, LOKACIJA}]` *i* `TEČAJ[{LOKACIJA, OIB_SUDIONIKA}]`. *Dobivamo relaciju danu tablicom 3.10 koja ima veću kardinalnost nego promatrana relacija* `TEČAJ` *iz čega slijedi da relacija* `TEČAJ` *ne zadovoljava JD pod 2.*

Pogledamo li JD pod 3 zaključujemo da je ona trivijalno zadovoljena.

Na kraju provjerimo je li JD pod 4 zadovoljena. U tu svrhu prvo provodimo prirodni spoj projekcija `TEČAJ[{NAZIV, LOKACIJA}]` *i* `TEČAJ[{NAZIV, OIB_SUDIONIKA}]` *te dobivamo relaciju danu tablicom 3.11. Zatim provodimo i prirodni spoj novodobivene relacije i projekcije* `TEČAJ[{LOKACIJA, OIB_SUDIONIKA}]` *te dobivamo relaciju kao u tablici 3.12. Primijetimo da je relacija dana tablicom 3.12 jednaka kao i promatrana relacija* `TEČAJ`, *pa zaključujemo da relacija* `TEČAJ` *zadovoljava promatranu JD.*

Spomenuli smo već u poglavlju 2 da je cilj normalizacije rastaviti polaznu relaciju u nekoliko manjih relacija kako bi se uklonila redundantnost podataka te kako bi se spriječile anomalije prilikom administriranja baze. To rastavljanje odgovara pojmu dekompozicije relacije koji ćemo sada i definirati.

Definicija 3.2.7. *Neka je $n \in \mathbb{N}, n \geq 2$, neka je R relacija sa skupom atributa A te neka su $A_1, A_2, \ldots, A_n$ podskupovi od A. Skup $D = \{R[A_1], R_2[A_2], \ldots, R[A_n]\}$ nazivamo* ***dekompozicijom relacije R*** *ako vrijedi:*

- *$A_i \neq \emptyset$, za svaki $i \in \{1, \ldots, n\}$ te*
- *$A_1 \cup A_2 \cup \ldots \cup A_n = A$.*

Kažemo da je dekompozicija relacije R ***trivijalna*** *ako je $R(A) \in D$.*

Primijetimo da su sve dekompozicije relacije stupnja jedan trivijalne.

Definicija 3.2.8. *Kažemo da su dvije relacije* ***jednake ili identične*** *ako im se podudara skup atributa te su im pripadne n-torke jednake.*

Kao što smo već i prije rekli, prilikom normalizacije željeli bismo polaznu relaciju dekomponirati u više relacija. Pri tome moramo biti svjesni da dekomponiranjem relacije možemo narušiti konzistentnost njezinih podataka.

Mogli bismo reći da je dekompozicija relacije *kvalitetna* ako primjenom prirodnog spoja nad svim relacijama iz dekompozicije dobivamo relaciju identičnu polaznoj. Drugim riječima, dekompozicija $D = \{R[A_1], R[A_2], \ldots, R[A_n]\}$ relacije R zadovoljava JD $\bowtie\{A_1, A_2, \ldots, A_n\}$. Navedeno upravo odgovara pojmu *spojne dekompozicije relacije bez gubitaka* koji ćemo sada i definirati.

Definicija 3.2.9. *Neka je $n \in \mathbb{N}, n \geq 2$, neka je R relacija sa skupom atributa A te neka su $A_1, A_2, \ldots, A_n$ podskupovi od A. Neka je $D = \{R[A_1], R_2[A_2], \ldots, R[A_n]\}$ dekompozicija relacije R. Skup D nazivamo* ***spojnom dekompozicijom bez gubitaka*** *(engl. lossless join decomposition)* ***relacije R*** *ako R zadovoljava JD $\bowtie\{A_1, A_2, \ldots, A_n\}$.*

Dekompoziciju relacija koja ne zadovoljava prethodni uvjet nazivamo ***spojnom dekompozicijom s gubitcima*** *(engl. lossy join decomposition)* ***relacije R****.*

Primjer 3.2.10. *Promotrimo li ponovno primjer 3.2.6 uočavamo da su dekompozicije:*

- $D_1 = \{\mathit{TEČAJ}[\{\mathit{NAZIV}\}], \mathit{TEČAJ}[\{\mathit{LOKACIJA}\}], \mathit{TEČAJ}[\{\mathit{OIB\_SUDIONIKA}\}]\}$
- $D_2 = \{\mathit{TEČAJ}[\{\mathit{NAZIV}, \mathit{LOKACIJA}\}], \mathit{TEČAJ}[\{\mathit{LOKACIJA}, \mathit{OIB\_SUDIONIKA}\}]\}$

primjeri spojnih dekompozicija s gubitcima, dok su

- $D_3 = \{\mathit{TEČAJ}[\{\mathit{NAZIV}, \mathit{LOKACIJA}, \mathit{OIB\_SUDIONIKA}\}], \mathit{TEČAJ}[\{\mathit{OIB\_SUDIONIKA}\}]\}$
- $D_4 = \{\mathit{TEČAJ}[\{\mathit{NAZIV}, \mathit{LOKACIJA}\}], \mathit{TEČAJ}[\{\mathit{NAZIV}, \mathit{OIB\_SUDIONIKA}\}], \mathit{TEČAJ}[\{\mathit{LOKACIJA}, \mathit{OIB\_SUDIONIKA}\}]\}$

primjeri spojnih dekompozicija bez gubitaka relacije TEČAJ.

Definicija pete normalne forme

Kao što je ranije spomenuto, pomoću pojmova dekompozicije i prirodnog spoja možemo definirati petu normalnu. Ideja je da se redundantnost podataka iz relacije izbjegne pomoću spojne dekompozicije bez gubitaka. Prije nego damo definiciju pete normalne forme bit će potrebno još uvesti pojam impliciranosti spojne ovisnosti relacije kandidatima za ključ.

Definicija 3.2.11. *Neka je $k \in \mathbb{N}, k \geq 2$, $C_1, \ldots, C_k$ skupovi atributa takvi da je $A = C_1 \cup \ldots \cup C_k$ te neka relacija R zadovoljava JD $\bowtie\{C_1, \ldots, C_k\}$. Kažemo da je JD $\bowtie\{C_1, \ldots, C_k\}$* ***implicirana kandidatima za ključ*** *ako svaki C_i predstavlja nadključ relacije R.*

Primijetimo da u definiciji 3.2.11 zapravo zahtijevamo da svaki skup atributa C_i je nadskup nekog kandidata za ključ.

Definicija 3.2.12. *Relacija R je u* ***petoj normalnoj formi (5NF)*** *ili* ***spojnoj normalnoj formi*** *(engl. project-join normal form, PJ/NF) ako je svaka netrivijalna spojna ovisnost koju relacija R zadovoljava implicirana kandidatima za ključ relacije R.*

Znamo da relacija zadovoljava neku JD ako odgovara prirodnom spoju svojih projekcija sa skupovima atributa iz dane JD. Stoga, možemo zaključiti da svaka netrivijalna JD koju relacija R u 5NF zadovoljava implicira spojnu dekompoziciju bez gubitaka u kojoj svaki element, svaka projekcija relacije R, kao skup atributa ima nadključ relacije R.

Propozicija 3.2.13. *Ako relacija zadovoljava 5NF, tada zadovoljava i 4NF.*

Dokaz. Ideja dokaza je karakterizacija višeznačne ovisnosti pomoću spojne ovisnosti, tj. relacija R zadovoljava MVD $A \twoheadrightarrow B$ ($A \twoheadrightarrow C$) ako i samo ako zadovoljava JD $\bowtie\{A \cup B, A \cup C\}$. Za više detalja pogledati u [5]. $\square$

Primjer 3.2.14. *Pretpostavimo da imamo relaciju*

`DDP(ID_DOB,ID_DIJELA,ID_PROJEKTA)`

danu tablicom 3.13 koja odražava činjenicu sljedećeg poslovnog pravila:
Ako (a) dobavljač s isporučuje dio p i (b) dio p se isporučuje projektu j i (c) projekt j se opskrbljuje od dobavljača s, tada (d) dobavljač s isporučuje dio p projektu j.

ID_DOB	ID_DIJELA	ID_PROJEKTA
s1	p1	j2
s2	p1	j1
s1	p2	j1
s1	p1	j1

Tablica 3.13: Relacija DDP

Primjerice, ako:

(a) *dobavljač s1 isporučuje dio p1 nekom projektu i*

(b) *netko isporučuje dio p1 projektu j1 i*

(c) *projekt j1 se opskrbljuje od dobavljača s1, tada*

(d) *dobavljač s1 isporučuje dio p1 projektu j1.*

Prvo, primijetimo, da uvjeti (a)–(d) odgovaraju redom retcima u tablici 3.13. Nadalje, relacija DDP ima samo jedan kandidat za ključ kojeg čine svi atributi relacije te zadovoljava spojnu ovisnost

$\bowtie\{\{ID\_DOB, ID\_DIJELA\}, \{ID\_DIJELA, ID\_PROJEKTA\}, \{ID\_PROJEKTA, ID\_DOB\}\}$

koja proizlazi iz ranije navedenog poslovnog pravila.

Uočimo da navedena JD nije implicirana kandidatima za ključ pa stoga dana relacija ne zadovoljava 5NF.

Dakle, relacija DDP ne zadovoljava 5NF, no njezinim dekomponiranjem na sljedeće relacije (njene projekcije):

- $DDP_1(ID\_DOB, ID\_DIJELA)$
- $DDP_2(ID\_DIJELA, ID\_PROJEKTA)$

- $\mathtt{DDP}_3(\mathtt{ID\_PROJEKTA}, \mathtt{ID\_DOB})$

dobivamo tri relacije koje zadovoljavaju 5NF jer se u njima ne nailazi na netrivijalne JD.

3.3 Šesta normalna forma

Šesta normalna forma predstavlja posljednji korak u normalizaciji podataka. Pri njoj se relacije dekomponira do razine kada više nije moguće spojno dekomponiranje bez gubitaka relacije (osim trivijalnog).

Definicija 3.3.1. *Relacija R je u* ***šestoj normalnoj formi (6NF)*** *ako je svaka spojna ovisnost koju relacija R zadovoljava trivijalna.*

Propozicija 3.3.2. *Ako relacija zadovoljava 6NF, tada zadovoljava i 5NF.*

Dokaz. Neka se relacija R nalazi u 6NF. Tada relacija R jedino zadovoljava trivijalne spojne ovisnosti, tj. ne postoji netrivijalna spojna ovisnost koju će R zadovoljavati. Dakle, relacija R se nalazi u 5NF. □

U sljedećem primjeru promotrit ćemo relaciju koja se ne nalazi u 6NF.

Primjer 3.3.3. *Neka relacija* `KVIZ(OIB_NATJECATELJA, IME, PREZIME, POSTOTAK)` *bilježi podatke o natjecateljima u nekom kvizu. Prilikom odigravanja kviza natjecatelj je dužan upisati svoje osobne podatke. Atribut* `POSTOTAK` *predstavlja postotak riješenosti kviza danog natjecatelja. Svaki natjecatelj ima pravo samo jedanput rješavati kviz.*

Dakle, podaci o kvizu se mogu bilježiti upisujući u tablicu 3.14.

OIB_NATJECATELJA	IME	PREZIME	POSTOTAK
535512179	Isaac	Newton	83
742871341	Carl Friedrich	Gauss	71
939851315	Bernhard	Riemann	100
467746345	Leonhard	Euler	95
223174582	Maria	Agnesi	95
523574558	Maria	Ardinghelli	90

Tablica 3.14: Relacija `KVIZ` koja nije u 6NF

Primijetimo da relacija `KVIZ` *nije u 6NF jer zadovoljava netrivijalnu JD*

$$\bowtie\{\{\mathtt{OIB\_NATJECATELJA,\ IME,\ PREZIME}\}, \{\mathtt{OIB\_NATJECATELJA,\ POSTOTAK}\}\}$$

impliciranu kandidatom za ključ $\{\mathtt{OIB\_NATJECATELJA}\}$.

Razdvajanjem relacije `KVIZ` *na relacije:*

- *KVIZ_IME(OIB_NATJECATELJA, IME)*
- *KVIZ_PREZIME(OIB_NATJECATELJA, PREZIME)*
- *KVIZ_POSTOTAK(OIB_NATJECATELJA, POSTOTAK)*

dobivamo relacije u 6NF. Ovakvom dekomponiranju relacije ćemo rijetko kada pristupiti u praksi, osim ako se radi o temporalnim bazama podataka (za više detalja pogledati poglavlje 23 u [5]).

3.4 Ostale normalne forme (EKNF, ETNF, DKNF)

Prije nego što krenemo na ostale normalne forme, važno je napomenuti da dodatno kroz ovaj dio rada pretpostavljamo da shema relacije sadrži i *ograničenja*.

Definicija 3.4.1. *Neka je X konačan skup atributa.* ***X-ograničenje*** *(ili ograničenje ako je X očit) je preslikavanje sa skupa svih relacija sa skupom atributa X u skup {TRUE, FALSE}.*

Neka je R relacija nad skupom atributa X. Kažemo da ***relacija R zadovoljava X-ograničenje*** *ili da X-ograničenje vrijedi za relaciju R ako je vrijednost X-ograničenja za relaciju R jednaka TRUE.*

Kažemo da ***relacija R ne zadovoljava X-ograničenje*** *ili da X-ograničenje ne vrijedi za relaciju R ako je vrijednost X-ograničenja za relaciju R jednaka FALSE.*

Već prije smo se susretali s ograničenjima poput FD, MVD i JD, samo ih nismo eksplicitno navodili kao ograničenja relacije. Naime, relacija zadovoljava ograničenje FD, MVD i JD ako je odgovarajuća (s obzirom na skup atributa) ovisnost prisutna. Na sljedećem primjeru promotrimo dane ovisnosti u svijetlu ograničenja.

Primjer 3.4.2. *Neka je dana relacija R(A, B, C).*

Promotrimo sljedeća ograničenja:

1. $A \to B$
2. $A \twoheadrightarrow B$
3. $\bowtie\{\{A, B\}, \{B, C\}\}$

Relacija R zadovoljava ograničenje FD pod 1. ako svaki njezin par n-torki koji ima jednaku A-vrijednost, ima jednaku i B-vrijednost. Slično, R zadovoljava ograničenje MVD pod 2. ako kada god u njoj postoje n-torke t_1 i t_2 koje imaju jednaku A-vrijednost, postoji i n-torka ($t_1(A), t_1(B), t_2(C)$). Relacija R će zadovoljiti ograničenje JD pod 3. ako je jednaka prirodnom spoju svojih projekcija R[{A,B}] i R[{B,C}].

Promotrimo sada primjere nekih drugih ograničenja nad relacijom.

Primjer 3.4.3. *Neka je dana relacija* `R(A, B, C)` *pri čemu domene atributa* `A`*,* `B` *i* `C` *odgovaraju skupu* $\mathbb{N}_+$.

Promotrimo sljedeća ograničenja:

1. $(\exists t_1 \in R)(t_1(A) = 1 \land t_1(B) = 1 \land t_1(C) = 1)$
2. $(\exists t_1, t_2, t_3 \in R)(t_1(A) = t_2(A) = t_3(A))$
3. $(\forall t_1 \in R)(t_1(C) \neq 2)$

Prvo ograničenje na relaciju `R` *postavlja uvjet da n-torka (1, 1, 1) mora biti sadržana u njoj, a drugo ograničenje da moraju postojati tri n-torke koje imaju jednaku* `A`*-vrijednost. Posljednje ograničenje traži da* `C`*-vrijednost svih n-torki u relaciji mora biti različita od 2.*

Normalna forma elementarnog ključa (EKNF)

Normalna forma elementarnog ključa nalazi se točno između 3NF i BCNF. Autor u [9] navodi da EKNF integrira kvalitete 3NF i BCNF izbjegavajući pritom probleme obiju normalnih formi.

Prije nego što definiramo EKNF bit će potrebno uvesti pojam elementarnog ključa.

Definicija 3.4.4. *Neka je R relacija i X kandidat za ključ relacije R. Kažemo da je X* ***elementaran ključ*** *ako postoji atribut A relacije R takav da je FD* $X \rightarrow A$ *netrivijalna i za svaki* $X' \subsetneq X$ *ne vrijedi FD* $X' \rightarrow A$.

Definicija 3.4.5. *Relacija R zadovoljava* ***normalnu formu elementarnog ključa (EKNF)*** *(engl. elementary key normal form) ako za svaku netrivijalnu FD* $X \rightarrow Y$ *vrijedi:*

- *X je nadključ relacije R ili*
- *Y podskup nekog elementarnog ključa*

Vidimo da je odmah iz definicije 3.4.5 jasno da BCNF povlači EKNF. Kako bismo pokazali da i EKNF povlači 3NF, dokazujemo sljedeću propoziciju.

Propozicija 3.4.6. *Ako relacija zadovoljava EKNF, tada zadovoljava i 3NF.*

Dokaz. Neka relacija R zadovoljava EKNF te neka je $X \rightarrow Y$ netrivijalna FD relacije R. Ako je X nadključ relacije R, dokaz je gotov. Stoga pretpostavimo da vrijedi drugi uvjet iz definicije 3.4.5, tj. da je Y podskup nekog elementarnog ključa K. Budući da je K elementarni ključ, on je po definiciji i kandidat za ključ. Dakle, Y je podskup nekog kandidata za ključ, odnosno svi elementi skupa Y su primarni atributi. □

Dakle, vrijedi BCNF ⇒ EKNF ⇒ 3NF. U primjerima koji slijede pokazat ćemo da obrati navedenih tvrdnji ne vrijede. Naime, opisat ćemo relaciju koja se nalazi u 3NF, no ne i u EKNF, a zatim i relaciju koja se nalazi u EKNF, no ne i u BCNF. Također, u primjerima ćemo za dane kandidate za ključ odrediti zadovoljavaju li oni i definiciju elementarnog ključa.

Primjer 3.4.7. *Prisjetimo se početne relacije* `STUDENT_MENTOR` *iz primjera 2.2.2. Pokazali smo već tada da dana relacija zadovoljava 3NF, no ne i BCNF. Provjerimo je li ona u EKNF.*

Relacija `STUDENT_MENTOR(JMBAG, OIB_STUDENTA, OIB_MENTORA)` *ima dva kandidata za ključ:* {`JMBAG, OIB_MENTORA`} *te* {`OIB_STUDENTA, OIB_MENTORA`}. *Primijetimo da dani kandidati za ključ nisu elementarni ključevi jer vrijede netrivijalne funkcionalne ovisnosti* `JMBAG` → `OIB_STUDENTA` *i* `OIB_STUDENTA` → `JMBAG`. *Zaključujemo da relacija* `STUDENT_MENTOR` *ne zadovoljava EKNF.*

Primjer 3.4.8. *Neka je dana relacija* `TEL(POZIVNI_BROJ, BROJ_TEL, GRAD)` *u kojoj se pohranjuju fiksni telefonski brojevi zajedno sa pozivnim brojem te gradom kojem pripadaju. Više gradova može imati isti pozivni broj (ovisno o županiji kojoj pripadaju), ali svaki grad ima samo jedan pozivni broj.*

Na temelju opisa možemo zaključiti da su u relaciji `TEL` *prisutne sljedeće funkcionalne ovisnosti:*

$$\{\mathtt{POZIVNI\_BROJ,\ BROJ\_TEL}\} \to \mathtt{GRAD} \tag{3.1}$$

$$\mathtt{GRAD} \to \mathtt{POZIVNI\_BROJ} \tag{3.2}$$

Također, dana relacija ima dva kandidata za ključ: $K_1 = \{\mathtt{POZIVNI\_BROJ,\ BROJ\_TEL}\}$ *te* $K_2 = \{\mathtt{GRAD,\ BROJ\_TEL}\}$, *iz čega pak zaključujemo da relacija zadovoljava 3NF (jer su svi atributi primarni). Nadalje, zbog funkcionalne ovisnosti* (3.1) *i činjenice da ne postoji podskup kandidata za ključ* K_1 *koji bi jedinstveno određivao* `GRAD`, *slijedi da je* K_1 *elementaran ključ. S druge strane, iz* $K_2 \to$ `POZIVNI_BROJ` *i FD* (3.2) *slijedi da* K_2 *nije elementaran ključ.*

Odredimo sada zadovoljava li promatrana relacija EKNF te BCNF. Primijetimo prvo da funkcionalna ovisnost (3.1) *ne narušava ni definiciju EKNF ni definiciju BCNF. Stoga je dovoljno promotriti funkcionalnu ovisnost* (3.2). *Budući da je* {`POZIVNI_BROJ`} *podskup elementarnog ključa* K_1 *slijedi da je relacija* `TEL` *u EKNF. No,* {`GRAD`} *nije nadključ relacije* `TEL`, *pa slijedi da relacija* `TEL` *ne zadovoljava BCNF.*

Vratimo se sada na uvod te obrazložimo [9] zašto BCNF nije uvijek bolje rješenje od 3NF kada relacijsku shemu promatramo u okvirima funkcionalnih ovisnosti.

Promotrimo ponovno primjer 3.4.8 i transformirajmo relaciju `TEL(POZIVNI_BROJ, BROJ_TEL, GRAD)` iz 3NF u BCNF slijedeći transformaciju iz potpoglavlja 2.2. Zbog funkcionalne ovisnosti (3.2) stvaramo novu relaciju `TEL2` s nadključem {`GRAD`}. U skup atributa relacije `TEL2` dodajemo i atribut `POZIVNI_BROJ`, a isti uklanjamo iz skupa atributa relacije `TEL`. Dakle, transformacijom dobivamo relacije `TEL(BROJ_TEL, GRAD)` i `TEL2(GRAD, POZIVNI_BROJ)` koje zadovoljavaju BCNF. Proučimo li pažljivo novodobivene relacije `TEL` i `TEL2` u BCNF, možemo primijetiti da smo transformacijom relacije `TEL` iz 3NF u BCNF izgubili *eksplicitnu* funkcionalnu ovisnost (3.1). Naime, relacija `TEL(BROJ_TEL, GRAD)` je relacija u kojoj je kandidat za ključ jednak skupu svih atributa pa u njoj više ne nailazimo na netrivijalne funkcionalne ovisnosti. U relaciji `TEL2(GRAD, POZIVNI_BROJ)` je pak prisutna samo netrivijalna funkcionalna ovisnost (3.2). Dakle, funkcionalna ovisnost (3.1) nije eksplicitno više prisutna u niti jednoj od novodobivenih relacija.

Normalna forma esencijalne n-torke (ETNF)

Dugo se mislilo da relacija mora biti u 5NF kako bi bila neredundantna, odnosno da će tek relacija u 5NF posjedovati svojstvo da ne sadrži informaciju o istom podatku, o istoj n-torci više puta.

U ovome poglavlju ćemo pokazati da postoji normalna forma ETNF, koja je slabija od 5NF, ali podjednako efikasna kao i 5NF u uklanjanju redundantnih n-torki. Autori u [3] tvrde da je ETNF nužna i dovoljna za uklanjanje redundantnosti te ju stoga preferiraju nad 5NF. Cilj ovog potpoglavlja bit će definirati što za n-torku znači da je neredundantna, tj. da je n-torka *esencijalna* te definirati ETNF u kojoj je svaka n-torka esencijalna.

Kao motivaciju za uvođenjem ETNF najprije ćemo u primjeru 3.4.11 promotriti relaciju sadrži redundantne n-torke i koja nije u 5NF. Uočit ćemo da 5NF rješava problem redundantnosti n-torki, no ne i najbolje, što ćemo i prikazati u primjeru 3.4.12.

Prije nego što pokažemo probleme koje uzrokuje redundantnost n-torki, uvedimo pojam *validne instance* relacijske sheme.

Definicija 3.4.9. *Kažemo da je relacija* ***validna instanca relacijske sheme*** *ako ima jednake atribute kao i shema te zadovoljava svako ograničenje sheme. U suprotnome, kažemo da relacija* ***nije validna instanca relacijske sheme****.*

Napomena 3.4.10. *Važno je napomenuti da se pojmom kandidata za ključ relacijske sheme referiramo na pojam kandidata za ključ validne instance promatrane sheme.*

Kada kažemo da je shema relacije redundantna podrazumijevamo da postoji njezina validna instanca koja ima redundantne n-torke.

Primjer 3.4.11. *Po uzoru na primjer 3.2.14, promotrimo relacijsku shemu* `DDP(ID_DOB,ID_DIJELA,ID_PROJEKTA)` *koja ima jednog kandidata za ključ koji je jednak skupu svih njezinih atributa te koja ima sljedeće ograničenje, odnosno netrivijalnu spojnu ovisnost:*

$$\bowtie\{\{ID\_DOB, ID\_DIJELA\}, \{ID\_DIJELA, ID\_PROJEKTA\}, \{ID\_PROJEKTA, ID\_DOB\}\} \quad (3.3)$$

Također kao i prije primijetimo da dana relacijska shema nije u 5NF jer spojna ovisnost (3.3) *nije implicirana kandidatima za ključ.*

Nadalje, lako se primjećuje da shema relacije `DDP` *redundantna. Neka je relacija R validna instanca od* `DDP` *te neka R sadrži barem sljedeće n-torke: (s, p, j'), (s', p, j) te (s, p', j) takve da je* $s \neq s'$, $p \neq p'$ *i* $j \neq j'$. *Tada zbog navedene spojne ovisnosti* (3.3) *(odnosno poslovnog pravila koje je objašnjeno u primjeru 3.2.14), relacija R sadrži i n-torku (s, p, j).*

Dakle, u relaciji R n-torka (s, p, j) je zapravo reprezentirana dva puta: prvi put eksplicitno sama sobom, a drugi put implicitno pomoću n-torki (s, p, j'), (s', p, j) i (s, p', j) i navedene JD.

Intuitivno, n-torka (s, p, j) je u potpunosti redundantna, što ćemo preciznije i definirati kasnije (vidi definiciju 3.4.16). Kako bismo riješili problem redundantnosti (relacije R), možemo posegnuti za dekompozicijom sheme `DDP` *sukladno JD* (3.3) *te tako dobiti neredundantne relacijske sheme* $DDP_1(ID\_DOB, ID\_DIJELA)$, $DDP_2(ID\_DIJELA, ID\_PROJEKTA)$ *i* $DDP_3(ID\_PROJEKTA, ID\_DOB)$ *koje se nalaze u 5NF.*

Primjer 3.4.12. *Promotrimo sada po uzoru na primjer 3.4.11 relacijsku shemu* `DDP'(ID_DOB,ID_DIJELA,ID_PROJEKTA)` *i pretpostavimo da su njezina ograničenja dana sa netrivijalnom spojnu ovisnošću* (3.3) *i netrivijalnom funkcionalnom ovisnošću*

$$\{ID\_DOB, ID\_DIJELA\} \to ID\_PROJEKTA \quad (3.4)$$

Iz navedenog slijedi da `DDP'` *ima jedan kandidat za ključ,* $\{ID\_DOB, ID\_DIJELA\}$.

Možemo reći da nam funkcionalna ovisnost intuitivno govori da dobavljači dostavljaju neki dio najviše jednom projektu.

U kojoj normalnoj formi se nalazi `DDP'`*? Budući da skup ograničenja ne sadrži netrivijalne višeznačnih ovisnosti, možemo zaključiti da relacija* `DDP'` *sigurno zadovoljava 4NF, no ne zadovoljava 5NF iz istog razloga kao i u primjeru 3.4.11.*

Slično kao i prije, neka je relacija R' validna instanca sheme `DDP'` *te neka relacija R' sadrži barem n-torke (s, p, j'), (s', p, j) i (s, p', j) takve da je* $s \neq s'$ *te* $p \neq p'$. *Primijetimo da sada zbog funkcionalne ovisnosti* (3.4) *ne možemo pretpostaviti da je* $j \neq j'$. *Naime, zbog JD* (3.3), *kao i u primjeru 3.4.11, slijedi da se n-torka (s, p, j) mora nalaziti u R'. No, funkcionalna ovisnost* (3.4) *povlači* $j = j'$

Prema tome, više ne nailazimo na redundantne n-torke jer je informacija koju sadrži n-torka (s,p,j) dana samo jedanput, tj. nije predstavljena implicitno pomoću n-torki (s,p,j'), (s',p,j) i (s, p',j). Naime, sada je ona jedna od njih.

Možemo zaključiti da 5NF nije nužna kako bi relacijska shema (relacija) bila neredundantna.

U daljnjem tekstu prvo ćemo definirati dva pojma usko vezana uz redundantnost n-torki, a zatim ćemo definirati i ETNF. Potpoglavlje ćemo zaključiti iskazom karakterizacije ETNF pomoću BCNF i JD te teoremom o hijerarhiji ETNF-a koji tvrdi da ako relacija zadovoljava ETNF, tada ona zadovoljava i 4NF, no ne nužno i 5NF.

Definicija 3.4.13. *Neka je R* relacijska shema, R validna instanca od R* i t n-torka u R. n-torka t je* ***parcijalno redundantna*** *u R ako vrijedi:*

- *postoji n-torka t' u R takva da* $t \neq t'$,
- *postoji netrivijalna FD* $X \to A$ *od R* te*
- $t[X] = t'[X]$.

Pojam parcijalne redundantnosti n-torke intuitivno je vezan uz FD. Naime, ako je t n-torka i FD $X \to A$, tada X-vrijednost od t jedinstveno određuje neku A-vrijednost. Točnije, za parcijalno redundantnu n-torku t postojat će n-torka t' različita od nje, ali takva da i njezina X-vrijednost određuje identičnu A-vrijednost. Dakle, informacija o nekoj A-vrijednosti je dana i od n-torke t i od n-torke t'.

Primjer 3.4.14. *Kako bismo na primjeru pokazali parcijalno redundantnu n-torku promotrimo ponovno po uzoru na primjer 2.2.2 relacijsku shemu* `STUDENT_MENTOR(JMBAG, OIB_STUDENTA, OIB_MENTORA)`. *Prisjetimo se, dana shema sadrži kao ograničenja i netrivijalne funkcionalne ovisnosti* `JMBAG` → `OIB_STUDENTA` *te* `OIB_STUDENTA` → `JMBAG`.

Promotrimo sada n-torke validne instance R relacijske sheme `STUDENT_MENTOR` *koja je predstavljena tablicom 3.15. Vidimo da su n-torke*

- $t_1 = (742871340, 223174582, 939851312)$ *i*
- $t_2 = (742871340, 223174582, 467746340)$

relacije R parcijalno redundantne. Naime, vrijedi da su one međusobno različite, FD `JMBAG` → `OIB_STUDENTA` *je zadovoljena te je* $t_1[\{JMBAG\}] = t_2[\{JMBAG\}]$. *Isto smo mogli zaključiti i pomoću FD* `OIB_STUDENTA` → `JMBAG` *te jednakosti* $t_1[\{OIB\}] = t_2[\{OIB\}]$. *Nadalje, primijetimo još i da n-torka* $t_3 = (742873022, 279180938, 467746340)$ *nije parcijalno redundantna.*

JMBAG	OIB_STUDENTA	OIB_MENTORA
742871340	223174582	939851312
742871340	223174582	467746340
742873022	279180938	467746340

Tablica 3.15: Validna instanca relacijske sheme STUDENT_MENTOR koja sadrži parcijalno redundantne n-torke

Definicija 3.4.15. *Neka je R* relacijska shema, S skup n-torki, i t n-torka. Kažemo da t* ***logički slijedi iz*** *S ako svaka validna instanca sheme R* koja sadrži sve n-torke iz skupa S nužno sadrži i n-torku t.*

Uočimo da je u primjerima 3.4.11 i 3.4.12 n-torka $t = (s, p, j)$ logički slijedila iz skupa $S = \{(s, p, j'), (s', p, j), (s, p', j)\}$.

Definicija 3.4.16. *Neka je R* relacijska shema, R validna instanca od R*, i t element (n-torka) instance R. Kažemo da je n-torka t* ***u potpunosti redundantna*** *u R ako postoji skup S n-torki u R takav da t nije element S te t logički slijedi iz S.*

Intuitivno, reći ćemo da je neka n-torka t u potpunosti redundantna u relaciji ako logički slijedi iz drugih n-torki.

Primijetimo da smo primjer n-torke koja je u potpunosti redundantna već naveli u primjeru 3.4.11.

Definicija 3.4.17. *Neka je R* relacijska shema te neka je relacija R validna instanca od R*. Kažemo da je n-torka t relacije R* ***esencijalna*** *ako nije ni parcijalno ni u potpunosti redundantna u R.*

Definicija 3.4.18. *Kažemo da relacijska shema R* zadovoljava* ***normalnu formu esencijalne n-torke (ETNF)*** *(engl. essential tuple normal form) ako je svaka n-torka u svakoj validnoj instanci od R* esencijalna.*

Iako je prethodna definicija precizna i točna, ipak nam nije od velike praktične koristi jer nam ne pomaže pri određivanju zadovoljava li neka relacijska shema ETNF. Stoga, navodimo karakterizaciju [3] koja je korisna pri ispitivanju zadovoljava li neka relacijska shema ETNF.

Teorem 3.4.19. *Neka je R* relacijska shema, te neka skup ograničenja od R* sadrži samo funkcionalne i spojne ovisnosti. Tada je R* u ETNF ako i samo ako je R* u BCNF i u svakoj spojnoj ovisnosti* $\bowtie\{C_1, \dots, C_k\}$ *od R* postoji* C_i, $i \in \{1, \dots, k\}$, *koji je nadključ od R*.*

Iz teorema 3.4.19 slijedi da `DDP` iz primjera 3.4.11 ne zadovoljava ETNF, dok `DDP'` iz primjera 3.4.12 zadovoljava ETNF.

Teorem 3.4.20. *Ako relacijska shema zadovoljava 5NF, tada ona zadovoljava i ETNF.*
Ako relacijska shema zadovoljava ETNF, tada ona zadovoljava i 4NF.
Obrati navedenih tvrdnji ne vrijede.

Dokaz. Zbog složenosti dokaza pogledati u [3]. □

Normalna forma domenskog ključa (DKNF)

U ovome dijelu rada prvo ćemo pojasniti ograničenja *ovisnost domene* i *ovisnost ključa* koji će nam biti potrebni prilikom definiranja DKNF. Zatim ćemo definirati i *anomalije* nastale prilikom umetanja ili brisanja podataka te što za relacijsku shemu znači da je *ispunjiva*. Na kraju pomoću prethodno navedenih pojmova izlažemo i karakterizaciju DKNF te smještamo DKNF između 5NF i 6NF.

Napomena 3.4.21. *U ovom dijelu rada pretpostavljamo da svaka validna instanca relacijske sheme zadovoljava 1NF.*

Sada, kao što smo i ranije spomenuli, kako bismo definirali DKNF bit će potrebno uvesti *ovisnost domene* i *ovisnost ključa*.

Definicija 3.4.22. *Neka je R relacija, A atribut relacije R, i S skup. Kažemo da relacija R zadovoljava ograničenje* ***ovisnost domene (DD)*** *(engl. domain dependency), u oznaci* $IN(A, S)$*, ako je za svaku n-torku t relacije R A-vrijednost od t element skupa S.*

Primjer 3.4.23. *Neka je R relacija, A atribut relacije R, i S skup svih brojeva od 10,000 do 100,000. Tada će relacija R zadovoljavati* $IN(A, S)$ *ako za svaku n-torku t relacije R vrijedi:* $10,000 \leq t(A) \leq 100,000$.

Definicija 3.4.24. *Neka je R relacija i K skup atributa relacije R. Kažemo da relacija R zadovoljava ograničenje* ***ovisnost ključa (KD)*** *(engl. key dependency), u oznaci* $KEY(K)$*, ako je K nadključ relacije R.*

Ovisnosti koje će nam biti od posebnog interesa prilikom definiranja DKNF i koje ćemo koristi u svrhu ograničenja sheme relacije su: funkcionalne, višeznačne i spojne ovisnosti, zatim ovisnost domene te ovisnost ključa.

Prije nego što predstavimo definiciju anomalije nastale prilikom umetanja podataka (n-torke) potrebno je definirati *kompatibilnost* n-torke s relacijom. Intuitivno, n-torke su kompatibilne s relacijom ako se mogu umetnuti u relaciju, odnosno ako odgovaraju relaciji s obzirom na njezin skup atributa i njihove vrijednosti te ako u relaciji već ne postoji identična n-torka.

Definicija 3.4.25. *Neka je R* shema relacija sa skupom atributa* $\{A_1, A_2, \ldots, A_n\}$*, R validna instanca od R*. Neka je t proizvoljna n-torka koja nije n-torka relacije R. Kažemo da je n-torka t* ***kompatibilna*** *sa R ako vrijedi:*

- $t \in dom(A_1) \times dom(A_2) \times \ldots dom(A_n)$,
- $t(A_i) \in S$ *za svaku DD IN(*A_i*,S) od R* te*
- $t[K] \neq s[K]$ *za svaku ovisnost ključa KEY(K) od R* i za svaku n-torku s relacije R.*

Definicija 3.4.26. *Kažemo da relacijska shema R* posjeduje* ***anomaliju umetanja*** *ako postoji validna instanca R od R* te n-torka t kompatibilna sa R takva da* $R \cup \{t\}$ *(relacija nastala umetanjem t u R) nije validna instanca od R*.*

Prema definiciji 3.4.25 umetanjem kompatibilne n-torke u relaciju se ne narušava niti jedno ograničenje DD i KD, iz čega slijedi da instancu od R* koja nije validna nakon umetanja kompatibilne n-torke možemo jedino dobiti ako R* ima neka druga ograničenja povrh DD i KD (primjerice ako ima FD).

U sljedećem primjeru izlažemo relacijsku shemu koja posjeduje anomaliju umetanja uzrokovanu funkcionalnom ovisnošću.

Primjer 3.4.27. *Neka je dana relacijska shema R* s atributima* `ZAPOSLENIK`*,* `ODJEL` *i* `VODITELJ` *te sljedećim ograničenjima (gdje je CHAR(20) skup riječi duljine najviše 20 znakova):*

- *IN(*`ZAPOSLENIK`*, CHAR(20))*
- *IN(*`ODJEL`*, CHAR(20))*
- *IN(*`VODITELJ`*, CHAR(20))*
- *KEY(*`ZAPOSLENIK`*)*
- `ODJEL` $\rightarrow$ `VODITELJ`

Cilj nam je sada pokazati da R posjeduje anomaliju umetanja. Neka je R relacija dana tablicom 3.16. R ima jednake atribute kao i R* te zadovoljava sva ograničenja od R*. Dakle, R je validna instanca sheme R*. Neka je n-torka* $t = (Cauchy, Matematika, Euler)$*. Primijetimo da je n-torka t kompatibilna sa relacijom R.*

Ako n-torku t umetnemo u relaciju R, dobivamo relaciju iz tablice 3.17 koja nije validna instanca sheme R jer narušava FD* `ODJEL` $\rightarrow$ `VODITELJ`*. Intuitivno [6], umetanjem n-torke t nije jasno je li namjera bila umetnuti novi podatak o zaposleniku Cauchyju ili je namjera bila ažurirati voditelja odjela matematika.*

`ZAPOSLENIK`	`ODJEL`	`VODITELJ`
Hilbert	matematika	Gauss
Pitagora	matematika	Gauss

Tablica 3.16: Validna instanca sheme R*

`ZAPOSLENIK`	`ODJEL`	`VODITELJ`
Hilbert	matematika	Gauss
Pitagora	matematika	Gauss
Cauchy	matematika	Euler

Tablica 3.17: Instanca sheme R* koja nije validna

Slijedi definicija anomalije brisanja.

Definicija 3.4.28. *Kažemo da relacijska shema R* posjeduje* ***anomaliju brisanja*** *ako postoji validna instanca R od R* i n-torka t u R takva da relacija* $R \setminus \{t\}$ *(relacija nastala uklanjanjem t iz R) nije validna instanca od R*.*

Pogledajmo prvo jedan jednostavan primjer relacijske sheme sa samo jednim atributom koja posjeduje anomaliju brisanja kada god njena validna instanca nema barem dvije različite n-torke.

Primjer 3.4.29. *Neka je R* relacijska shema sa samo jednim atributom* `A` *te neka su ograničenja (gdje NAT predstavlja skup prirodnih brojeva) nad njom sljedeća:*

- $IN(\mathtt{A}, NAT)$
- $(\exists t_1, t_2)(t_1 \neq t_2)$

Uočimo da R posjeduje anomaliju brisanja što možemo dokazati uzimanjem validne instance čija je kardinalnost jednaka dva.*

Primjer 3.4.30. *Neka je R* relacijska shema koja se sastoji od atributa* `A`*,* `B` *i* `C` *te sljedećih ograničenja (gdje NAT predstavlja skup prirodnih brojeva):*

- *IN(*`A`*, NAT)*
- *IN(*`B`*, NAT)*
- *IN(*`C`*, NAT)*
- $\mathtt{A} \twoheadrightarrow \mathtt{B}$

Neka je R validna instanca sheme R dana tablicom 3.18. Za pokazati anomaliju brisanja u R* dovoljno je obrisati bilo koju n-torku relacije R i time će biti narušeno ograničenje višeznačne ovisnosti* $\mathtt{A} \twoheadrightarrow \mathtt{B}$*.*

Slijedi definicija DKNF.

A	B	C
0	0	0
0	0	1
0	1	0
0	1	1

Tablica 3.18: Validna instanca R sheme R*

Definicija 3.4.31. *Neka je R* relacijska shema. Kažemo da R* zadovoljava **normalnu formu domenskog ključa (DKNF)** (engl. domain key normal form) ako sva ograničenja od R* su ovisnosti domene i ovisnosti ključa.*

Kako bismo mogli dati karakterizaciju DKNF bit će potrebno uvesti pojam *ispunjivosti* relacijske sheme.

Definicija 3.4.32. *Relacijska shema R* je **ispunjiva** (engl. satisfiable) ako ima barem jednu validnu instancu.*

Relacijska shema će biti ispunjiva ako njezina ograničenja nisu međusobno kontradiktorna. Primjer neispunjive relacijske sheme je shema koja posjeduje ograničenje $\exists t(t \neq t)$ ili pak ograničenje poput DD $IN(A, \{1\})$ i $\forall t(t[A] = 2)$.

Bitno je i napomenuti da DD $IN(A, \emptyset)$ u skupu ograničenja neke relacijske sheme ne znači nužno da je ta shema neispunjiva, već samo povlači tvrdnju da joj je jedina validna instanca *prazna relacija* (relacija koja ne sadrži nijednu n-torku). S druge strane, svaka shema čiji se skup ograničenja sastoji samo od DD i KD je ispunjiva jer je prazna relacija (s odgovarajućim skupom atributa) njezina validna instanca.

Teorem koji slijedi daje prvu karakterizaciju DKNF za ispunjivu relacijsku shemu.

Teorem 3.4.33. *Ispunjiva relacijska shema je u DKNF ako i samo ako nema anomaliju umetanja i nema anomaliju brisanja.*

Dokaz. Pogledati u [6]. □

Pokazali smo da shema u primjeru 3.4.27 posjeduje anomaliju umetanja, a jer DD, KD i FD neće biti narušena brisanjem n-torke iz relacije koja ih zadovoljava, vidimo da ista shema nema anomaliju brisanja. Također, pokazali smo da shema u primjeru 3.4.29 posjeduje anomaliju brisanja, a jer ograničenja njezine validne instance neće biti narušena umetanjem kompatibilne n-torke slijedi da ona nema anomaliju umetanja.

Postoji zanimljiva karakterizacija DKNF gdje je nepostojanje anomalije brisanja zamijenjeno drugim uvjetom.

Korolar 3.4.34. *Neka je R* ispunjiva relacijska shema. Tada R* zadovoljava DKNF ako i samo ako vrijedi:*

(1) R nema anomaliju umetanja te*

(2) prazna relacija (s odgovarajućim atributima) je validna instanca od R.*

Dokaz. Pretpostavimo prvo da R* zadovoljava DKNF. Prema teoremu 3.4.33 shema R* nema anomaliju umetanja. Iz definicije DKNF slijedi da skup ograničenja R* sadrži samo DD i KD, pa je prazna relacija njezina validna instanca.

Za drugi smjer dokaza, prema teoremu 3.4.33 dovoljno je dokazati da prazna relacija nema anomaliju umetanja (pogledati u [6]). □

Na kraju možemo primijetiti da će svaka prazna relacija (s odgovarajućim skupom atributa) zadovoljiti sva ograničenja proizašla iz DD, KD, FD, MVD i JD (sa identičnim atributima), no ona prema prethodnom korolaru neće nužno povlačiti da se svaka relacijska shema nalazi u DKNF.

Primjeri 3.4.27 i 3.4.30 vezani su uz sheme koje nisu u DKNF jer zadovoljavaju uvjet (2), ali ne i uvjet (1) korolara 3.4.34. Relacijska shema iz primjera 3.4.29 zadovoljava uvjet (1), ali ne i uvjet (2) korolara 3.4.34 pa stoga i ona nije u DKNF.

Iskažimo i posljednji teorem u ovome radu koji govori gdje se u hijerarhiji normalnih formi nalazi DKNF. Dokaz teorema izostavljamo zbog njegove složenosti te se isti može pogledati u [6].

Teorem 3.4.35. *Ako relacijska shema zadovoljava DKNF, tada zadovoljava i 5NF.*

Na kraju, primjerom pokažimo da relacijska shema koja zadovoljava DKNF, ne mora nužno zadovoljavati i 6NF.

Primjer 3.4.36. *Neka je dana relacijska shema `ZAP_CIJENA` s atributima `ID_ZAPOSLENIKA`, `ID_ODJELA` i `CIJENA_DANA` te sljedećim ograničenjima (gdje je NAT skup prirodnih brojeva, a S skup svih brojeva od 100 do 1000):*

- *IN(`ID_ZAPOSLENIKA`, NAT)*
- *IN(`ID_ODJELA`, NAT)*
- *IN(`CIJENA_DANA`, S)*
- *KEY(`ID_ZAPOSLENIKA`)*

Primijetimo prvo da iz definicije 3.4.31 slijedi da ZAP_CIJENA *zadovoljava DKNF.*

S druge strane, iz ograničenja $KEY(\texttt{ID\_ZAPOSLENIKA})$ *možemo izvesti i ograničenje JD:*

$\bowtie\{\{\texttt{ID\_ZAPOSLENIKA, ID\_ODJELA}\}, \{\texttt{ID\_ZAPOSLENIKA, CIJENA\_DANA}\}\}.$

Slijedi, ZAP_CIJENA *nije u 6NF.*

Bibliografija

[1] *Prezentacija za vježbe iz kolegija Baze podataka, Prirodoslovno-matematički fakultet–Matematički odsjek, Zagreb*, (2021.).

[2] E. F. Codd, *Further Normalization of the Data Base Relational Model*, Research Report / RJ / IBM / San Jose, California **RJ909** (1971.).

[3] H. Darwen, J. Date i R. Fagin, *A Normal Form for Preventing Redundant Tuples in Relational Databases*, ACM International Conference Proceeding Series (2012.).

[4] C. J. Date, *Database Design and Relational Theory, Normal Forms and All That Jazz*, Apress, 2019.

[5] C.J. Date, *An Introduction to Database Systems*, Addison-Wesley Longman Publishing Co., Inc., 2004.

[6] R. Fagin, *A normal form for relational database that is based on domains and keys*, ACM Transactions on Database Systems **6** (1981.), 387–415.

[7] R. Manger, *Baze podataka*, Element, 2012.

[8] T. Teorey, S. Light i Nadeau T., *Database Modeling and Design: Logical Design, Fourth Edition*, Morgan Kaufmann, 2005.

[9] C. Zaniolo, *A new normal form for the design of relational database schemata*, ACM Trans. Database Syst. **7** (1982.), 489–499.

Sažetak

U ovome radu proučavamo normalne forme u relacijskim bazama podataka pri čemu su nam od posebnog interesa više normalne forme.

U prvom poglavlju definiramo osnovne pojmove vezane uz relacijski model podataka. Uvodimo i pojam nadključa, kandidata za ključ te primarnog ključa relacije kao pojmove od interesa pri definiranju normalnih formi.

U drugom poglavlju bavimo se nižim normalnim formama. Na početku poglavlja definiramo i razmatramo prvu normalnu formu. Zatim, uvodimo pojam funkcionalne ovisnosti pomoću kojeg definiramo drugu, treću te Boyce-Coddovu normalnu formu. Na kraju poglavlja dajemo i usporedbu različitih definicija druge, odnosno treće normalne forme.

U trećem poglavlju promatramo više normalne forme u relacijskim bazama podataka. Definiramo višeznačnu ovisnost i četvrtu normalnu formu te spojnu ovisnost i petu, odnosno šestu normalnu formu. Treće poglavlje završavamo definiranjem normalne formu elementarnog ključa, normalne forme esencijalne n-torke te normalne forme domenskog ključa koje uspoređujemo s prethodno uvedenim normalnim formama.

Summary

In this thesis, we study normal forms in a relational databases where, of particular interest, are higher normal forms.

In the first chapter, we define the basic concepts related to the relational database model. We also introduce the concept of superkey, candidate key and primary key as concepts of interest when defining normal forms.

In the second chapter, we deal with lower normal forms. At the beginning of the chapter, we define and consider first normal form. Then, we introduce the concept of a functional dependency by which we define second, third and Boyce-Codd's normal form. At the end of the chapter, we also give a comparison of different definitions of second and third normal form.

In the third chapter, we observe higher normal forms in the relational databases. First, we define a multivalued dependency and fourth normal form. Then, we define a join dependency and fifth and sixth normal form, respectively. The third chapter ends by defining elementary key normal form, essential tuple normal form and domain key normal form, which we compare with the previously introduced normal forms.

Životopis

Rođena sam 24. veljače 1999. godine u Zagrebu. Osnovnu školu Sesvetski Kraljevec u Zagrebu završavam 2013. godine. Iste godine upisujem matematički smjer III. gimnazije u Zagrebu. Po završetku gimnazijskog obrazovanja 2017. godine upisujem preddiplomski studij Matematike na Prirodoslovno-matematičkom fakultetu u Zagrebu. Nakon završenog preddiplomskog studija 2021. godine upisujem diplomski studij Računarstvo i matematika na istom fakultetu.