

Projektovanje baza podataka

Logičko modeliranje – Prečiščavanje sheme

3. jun 2026.

Sadržaj

- 1 Funkcionalne zavisnosti
- 2 Osnovne normalne forme

Napomena

Slajdovi su nastali obradom slajdova prof. dr Saše Malkova za predmet Projektovanje baza podataka

Prečišćavanje sheme

- Podsetimo se, logički model se pravi iterativno:
 - 1. korak: prevođenje konceptualnog modela u logički
 - 2. korak: prečišćavanje sheme, iterativno
 - Svaka naredna iteracija se pravi menjanjem prethodne (engl. flexing)

Prevođenje konceptualnog modela u logički

- Primarni cilj prevođenja konceptualnog modela u relaciji:
 - Prilagođavanje modela domena konkretnom izabranom modelu podataka (npr. relacionom modelu)
 - Potpunost, tj. zadržavanje svih značajnih odnosa
- Rezultat prevođenja, međutim, često nije do kraja doteran
 - Potrebno je dodatno prečišćavanje da bi bio potpuno u skladu sa modelom podataka

Prečišćavanje sheme

- Pri prečišćavanju sheme
 - Analiziramo dobijeni logički model
 - Za sve pojedinačne elemente proveravamo usklađenost sa osnovnim konceptima odgovarajućeg modela podataka
 - Po potrebi uvodimo izmene u model
 - Vodimo računa o zahtevima (pre svega funkcionalnim)

Prečišćavanje sheme i relacioni model

- U slučaju relacionog modela pri prečišćavanju sheme se najviše bavimo onim aspektima projektovanja koji su od primarnog značaja za relacioni model
 - Eliminacija *redundantnosti*
 - Dosledna primena pravila integriteta
- Kao osnovno sredstvo koristimo *funkcionalne zavisnosti*

Problemi redundantnosti

- Redundantno čuvanje podataka
 - neki podaci se ponovljeno čuvaju, pa više kopija istog podatka nepotrebno opterećuje prostorne kapacitete
- Anomalije ažuriranja
 - ako se jedna kopija ponovljenog podatka ažurira, baza podataka će biti nekonzistentna ako se ne ažuriraju i ostale kopije
- Anomalije dodavanja
 - zapisivanje nekih podataka može da dovodi do nekonzistentnosti ako se ne zapišu još neki dodatni podaci
- Anomalije brisanja
 - brisanje nekih podataka može da dovodi do nekonzistentnosti ako se ne obrišu još neki podaci

Primer redundantnosti

- UpisanKurs(
broj indeksa,
ime i prezime studenta,
šifra predmeta,
naziv predmeta,
ime nastavnika)
- Postoji potencijalan problem sa redundantnim podacima o nastavnicima, studentima i predmetima

...

Broj indeksa	Ime i prezime studenta	Šifra predmeta	Naziv predmeta	Ime nastavnika
211/2015	Goran Petrović	M101	Analiza 1	Milan Marić
211/2015	Goran Petrović	M102	Linearna algebra	Petar Simić
212/2015	Tanja Mitrović	M101	Analiza 1	Milan Marić
212/2015	Tanja Mitrović	M102	Linearna algebra	Petar Simić

Tabela: Primer relacije UpisanKurs

Nedefinisane vrednosti i redundantnost

- Nedefinisane vrednosti mogu da pomognu u rešavanju nekih anomalija dodavanja i brisanja
 - Ako moramo da dodamo podatak a ne znamo druge potrebne podatke, onda možemo da navedemo nedefinisane vrednosti
 - npr. ako student upisuje kurs a ne znamo nastavnika, možemo da navedemo da je nastavnik nedefinisan
 - Ako hoćemo da obrišemo neke redundantne podatke, a da ne brišemo cele redove, onda možemo da umesto brisanja upišemo nedefinisane vrednosti
 - npr. ako brišemo podatak o nastavniku na kursu a ne želimo da obrišemo podatke o studentima koji su upisali kurs, možemo da upišemo da je nastavnik nedefinisan

Dekompozicija

- Redundantnost se pojavljuje na mestima na kojima postoje ***ne sasvim prirodne*** veze između atributa
- Uobičajeno i **ispravno** sredstvo za rešavanje redundantnosti je dekompozicija
- Ideja je da se jedna relacija sa mnogo atributa zameni većim brojem relacija sa po manje atributa

...

- Na primer, možemo da podelimo relaciju:
 - UpisanKurs(
broj indeksa,
ime i prezime studenta,
šifra predmeta,
naziv predmeta,
ime nastavnika)
- na relacije:
 - UpisanKurs(
broj indeksa,
ime i prezime studenta,
šifra predmeta,
naziv predmeta)
 - NastavnikKursa(
šifra predmeta,
ime nastavnika)
- (tako smo rešili samo deo problema)

Primer dekompozicije relacije

211/2015	Goran Petrović	M101	Analiza 1
211/2015	Goran Petrović	M102	Linearna algebra
212/2015	Tanja Mitrović	M101	Analiza 1
212/2015	Tanja Mitrović	M102	Linearna algebra

M101	Milan Marić
M102	Petar Simić

Pitanja u vezi dekompozicije

- Svaki put kada razmišljamo o dekompoziciji moramo da se zapitamo:
 - Koji problem želimo da rešimo predloženom dekompozicijom?
 - Da li ga zaista i rešavamo?
 - Da li tako proizvodimo neke nove probleme?
- Odgovore na ova pitanja daju
 - analiza **funkcionalnih zavisnosti**
 - teorijske osnove **normalnih formi** relacija

Sadržaj

- 1 Funkcionalne zavisnosti
- 2 Osnovne normalne forme

“Aksiome” izvođenja FZ

- Aksiome izvođenja FZ (nazivaju se i Armstrongove aksiome)
- Nazivaju se *aksiome izvođenja* zato što se sve složenije FZ mogu izvesti iz osnovnih FZ primenom aksioma izvođenja

“Aksiome” izvođenja FZ

- Za sve skupove atributa X, Y i Z neke relacije važe:

- **Refleksivnost**

$$X \rightarrow X$$

koristi se i u ekvivalentnom obliku

$$Y \subseteq X \Rightarrow X \rightarrow Y$$

- **Proširivost**

$$X \rightarrow Y \Rightarrow (\forall Z) XZ \rightarrow Y$$

koristi se i u ekvivalentnom obliku

$$X \rightarrow Y \Rightarrow (\forall Z) XZ \rightarrow YZ$$

- **Tranzitivnost**

$$X \rightarrow Y \wedge Y \rightarrow Z \Rightarrow X \rightarrow Z$$

Funkcionalne zavisnosti, primer

Indeks	Ime	Prezime	Šif.p.	Naziv predmeta	Ime nastavnika
211/2015	Goran	Petrović	M101	Analiza 1	Milan Marić
211/2015	Goran	Petrović	M102	Linearna algebra	Petar Simić
212/2015	Tanja	Mitrović	M101	Analiza 1	Milan Marić
212/2015	Tanja	Mitrović	M102	Linearna algebra	Jovana Antić
212/2015	Tanja	Mitrović	P100	Programiranje 1	Dragana Perić

● Primeri **refleksivnosti**:

- $\{\text{Indeks, Ime, Prezime, Naziv predmeta}\} \rightarrow \{\text{Prezime}\}$
- $\{\text{Indeks, Ime, Prezime, Naziv predmeta}\} \rightarrow \{\text{Indeks, Ime, Prezime}\}$
- $\{\text{Indeks, Ime, Prezime, Naziv predmeta}\} \rightarrow \{\text{Indeks, Naziv predmeta}\}$
- ... i druge

Funkcionalne zavisnosti, primer

Indeks	Ime	Prezime	Šif.p.	Naziv predmeta	Ime nastavnika
211/2015	Goran	Petrović	M101	Analiza 1	Milan Marić
211/2015	Goran	Petrović	M102	Linearna algebra	Petar Simić
212/2015	Tanja	Mitrović	M101	Analiza 1	Milan Marić
212/2015	Tanja	Mitrović	M102	Linearna algebra	Jovana Antić
212/2015	Tanja	Mitrović	P100	Programiranje 1	Dragana Perić

● Primeri proširivosti (na skupu torki):

- $\{\text{Ime}\} \rightarrow \{\text{Prezime}\} \Rightarrow \{\text{Ime}, \text{Indeks}\} \rightarrow \{\text{Prezime}, \text{Indeks}\}$
- $\{\text{Šif.p.}\} \rightarrow \{\text{Naziv predmeta}\} \Rightarrow \{\text{Šif.p.}, \text{Indeks}\} \rightarrow \{\text{Naziv predmeta}, \text{Indeks}\}$
- ... i druge

Funkcionalne zavisnosti, primer

Indeks	Ime	Prezime	Šif.p.	Naziv predmeta	Ime nastavnika
211/2015	Goran	Petrović	M101	Analiza 1	Milan Marić
211/2015	Goran	Petrović	M102	Linearna algebra	Petar Simić
212/2015	Tanja	Mitrović	M101	Analiza 1	Milan Marić
212/2015	Tanja	Mitrović	M102	Linearna algebra	Jovana Antić
212/2015	Tanja	Mitrović	P100	Programiranje 1	Dragana Perić

● Primeri **tranzitivnosti**:

- $\{\text{Ime}, \text{Šif.p.}\} \rightarrow \{\text{Prezime}, \text{Šif.p.}\},$
 $\{\text{Prezime}, \text{Šif.p.}\} \rightarrow \{\text{Ime nastavnika}\}$
 $\Rightarrow \{\text{Ime}, \text{Šif.p.}\} \rightarrow \{\text{Ime nastavnika}\}$
- ... i druge

Najvažnije izvedene osobine FZ

- Izvode se iz aksioma
- Često se i one nazivaju aksiomama
- Najvažnije su:
 - Aditivnost (ili unija)
 - Projektivnost (ili dekompozicija)
 - Pseudotranzitivnost

Aditivnost

- Aditivnost

$$X \rightarrow Y \wedge X \rightarrow Z \Rightarrow X \rightarrow YZ$$

Projektivnost

- **Projektivnost** ili dekompozicija

$$X \rightarrow YZ \Rightarrow X \rightarrow Y \wedge X \rightarrow Z$$

Pseudotranzitivnost

- Pseudotranzitivnost

$$X \rightarrow Y \wedge YZ \rightarrow W \Rightarrow XZ \rightarrow W$$

- Često se koristi u nešto osnovnijem obliku:

$$X \rightarrow Y \wedge XY \rightarrow Z \Rightarrow X \rightarrow Z$$

Značaj funkcionalnih zavisnosti

- Značaj FZ je višestruk, ali za nas je najvažnije da:
 - postojanje složenih FZ može da ukaže na redundantnosti
 - na primer, ako u relaciji ima tranzitivnih FZ, onda su ispunjeni preduslovi za pojavu redundantnog sadržaja relacije
 - postojanje FZ nam ukazuje na mogućnost da se relacija dekomponuje bez gubitka informacija
 - na primer, razdvajanjem tranzitivnih FZ u različite relacije

Funkc. zavisnosti i redundantnost

- Neka su X, Y, Z podskupovi atributa relacije R
- Neka važe FZ: $X \rightarrow Y$ i $Y \rightarrow Z$
 - (primetimo da onda važi i $X \rightarrow Z$)
- Ako bar jedna od ovih FZ nije 1-1, onda u relaciji R mogu da se pojave redundantne vrednosti
- Na primer, neka $X \rightarrow Y$ nije 1-1, onda relacija može da sadrži torke (projektovane na XYZ):
 - $(x1, y1, z1)$
 - $(x2, y1, z1)$
 - vrednosti atributa Z se redundantno ponavljaju i moraju da se menjaju kada god se menja Y

Funkc. zavisnosti i redundantnost

- U načelu, skoro svaka relacija u kojoj važi više funkcionalnih zavisnosti predstavlja mogućnost za pojavljivanje redundantnih torki
 - osim ako su sve te FZ 1-1
- Iz ugla redundantnosti, u idealnom slučaju bi u svakoj relaciji trebalo da postoji tačno jedna FZ
 - Uloga relacije bi onda bila da modelira tu jednu FZ
 - U praksi to baš ne može tako
 - Olakšava se ispravnost ažuriranja
 - Ali često mora da se ažurira više relacija
 - Opterećujuće iz ugla upita, zbog mnogo spajanja
- Da li možemo relaciju sa više FZ da podelimo na više relacija a da ne izgubimo informacije?

Zatvorenje skupa funkc. zavisnosti

- U razmatranju FZ često je potrebno sagledati međuzavisnosti između skupova zavisnosti ili između skupova atributa
- Neka je F neki skup FZ relacije R . **Zatvorenje od F** , u oznaci F^+ , je najmanji skup FZ koji sadrži F , takav da primenom Armstrongovih aksioma na F^+ ne može da se dobije nijedna FZ koja nije već u F^+ .
- Neka je F neki skup FZ relacije R i neka je X neki skup atributa te relacije. Neki skup atributa je **zatvorenje od X nad F** , u oznaci X^+ , akko postoji FZ $X \rightarrow X^+$ u F^+ i ako za svaku drugu FZ $X \rightarrow Z$ iz F^+ važi $Z \subseteq X^+$.
 - t.j. X^+ je maksimalan skup atributa zavisnih od X

Skup zavisnih atributa

- **Skup zavisnih atributa** X^+ nekog skupa atributa X je skup svih atributa koji su (posredno ili neposredno) funkcionalno zavisni od atributa iz X
 - t.j. predstavlja zatvorenje u odnosu na sve FZ u relaciji
- (Algoritme za konstrukciju zatvorenja FZ i atributa naučiti iz knjige GPL)

Funkcionalne zavisnosti i integritet

- Koncept integriteta u relacionom modelu je tesno povezan sa funkcionalnim zavisnostima
- Videćemo kako se ključ-kandidat, primarni ključ i jedinstveni ključ povezuju sa funkcionalnim zavisnostima

Natključ i ključ-kandidat

- Skup atributa X relacije R je natključ akko je $X^+ = \text{Attr}(R)$
 - t.j. neki skup atributa možemo da koristimo kao ključ za identifikovanje torki relacije akko su svi ostali atributi relacije (neposredno ili posredno) zavisni od njega
- Natključ X relacije R je ključ-kandidat akko za svaki drugi natključ Y važi: $Y \subseteq X \Rightarrow Y = X$.
 - t.j. neki natključ je ključ-kandidat ako ne postoji natključ koji je njegov pravi podskup
 - primetimo da može postojati i drugi natključ i drugi ključ kandidat koji nisu podskupovi od X

Integritet ključa

- Uslovu ključa odgovara funkcionalna zavisnost oblika:
 $\{ \text{atributi ključa} \} \rightarrow \{ \text{svi ostali atributi} \}$
- Primetimo da je ovo problematično ako ključ sadrži attribute koji mogu da imaju nedefinisane vrednosti
 - FZ se definišu na poređenju, tj.
 - ako su jednaki ključevi, onda su jednaki i ostali atributi
 - Kako se porede nedefinisane vrednosti?

Integritet jedinstvenosti entiteta

- Zbog toga se integritet jedinstvenosti entiteta definiše tako da predstavlja dodatno proširenje i formalizaciju integriteta ključa
- Za primarni ključ se bira ključ kandidat koji ne sme da ima nedefinisane vrednosti:
 - 1. { **atributi primarnog ključa** } $\rightarrow$ { **svi ostali atributi** }
 - 2. atributi primarnog ključa ne smeju da imaju nedefinisane vrednosti

Integritet jedinstvenog ključa

- Integritet jedinstvenog ključa je potencijalno slabiji oblik jedinstvenosti od integriteta primarnog ključa
- Imamo tri alternativna oblika:
 1. { **atributi jedinstvenog ključa** } $\rightarrow$ { **svi ostali atributi** }
 - 2a. atributi jedinstvenog ključa ne smeju da imaju nedefinisane vrednosti
 - isto kao i jedinstvenost entiteta
 - 2b. ako atributi jedinstvenog ključa imaju nedefinisane vrednosti, smatra se da su one međusobno identične
 - različiti redovi ne mogu da sadrži iste vrednosti definisanih atributa i iste kombinacije nedefinisanih atributa
 - 2v. ako atributi jedinstvenog ključa imaju nedefinisane vrednosti, smatra se da su one međusobno različite
 - različiti redovi mogu da sadrže iste vrednosti definisanih atributa i iste kombinacije nedefinisanih atributa
 - kao da se ne proverava jedinstvenost ključeva koji imaju nedefinisane attribute

Dekompozicije relacione sheme

- Dekompozicija relacione sheme je zamenjivanje relacione sheme dvema (ili više) relacionim shemama koje zajedno sadrže sve attribute polazne sheme
- Obično se govori o relacijama, a misli se na relacione sheme

Potpunost dekompozicije

- Dekompozicija je **potpuna** (kompletna) ako garantuje da će odgovarajuće relacije uvek sadržati sve one informacije koje bi sadržala polazna relacija
- Potpunosť se formalizuje preko **rekonstruktibilnosti**
 - Kažemo da je dekompozicija potpuna akko polazna relacija može da se rekonstruiše spajanjem rezultujućih relacija

Očuvanje zavisnosti

- Potpunost dekompozicije može da se svede na očuvanje FZ
 - Dekompozicija **čuva funkcionalne zavisnosti** ako se iz FZ na dekomponovanim relacijama mogu rekonstruisati sve FZ na polaznoj relaciji
- Teorema: **Ako dekompozicija čuva funkcionalne zavisnosti, onda je ona potpuna.**
- Predstavlja jedan od najvažnijih teorijskih elemenata relacionog modela i osnov za prečišćavanje sheme i eliminaciju redundantnosti iz modela

Funkc. zavisnosti na relacijama

- Teorema: Neka je $\text{Attr}(R)$ skup atributa relacije R i neka za neprazne poksкупove atributa $X, Y \subseteq \text{Attr}(R)$ ваži FZ $X \rightarrow Y$, tada ваži jednakost $R = R[XY] * R[XZ]$, где је $Z = \text{Attr}(R) \setminus XY$.
 - t.j. ako iz relacije izdvojimo novu relaciju tako da modelira jednu FZ, a pri tome u relaciji zadržimo attribute domena zavisnosti, onda je dekompozicija potpuna
 - ideja je da na osnovu nove relacije možemo da rekonstruišemo nedostajuće attribute

Da li se problemi usled redundantnosti vide u konceptualnom modelu?

- Neke zavisnosti nisu očigledne u konceptualnom modelu
- Posledica je da nakon prevođenja može da se dobije logički model u kome postoji redundantnost
- Na primer, možemo da dobijemo relacije:
Student(indeks, ime i prezime, nivo studija)
Smer(oznaka, naziv, trajanje)
koje su u odnosu:
Student—(*)—studira—(1)—Smer
(oznake kardinalnosti kao u UML-u)
- Sve zavisnosti u relacijama Student i Smer su ispravne...

...

- Redundantnost može da se uoči, ali je očiglednija kada se u model uvedu i ključevi vezanih relacija (ili se posmatra međusobna zavisnost odnosa i atributa):
 - Student(indeks, ime i prezime, nivo studija, **oznaka smer**a)
 - Smer(**oznaka**, naziv, trajanje)
- Sada je očigledno da postoji redundantnost atributa *nivo studija* u relaciji Student
 - Zbog zavisnosti: oznaka smer → nivo studija
- Ispravno rešenje bi bilo:
Student(indeks, ime i prezime, oznaka smer)
Smer(oznaka, naziv, nivo studija, trajanje)

Sadržaj

- 1 Funkcionalne zavisnosti
- 2 Osnovne normalne forme

Normalne forme

- **Normalne forme** su specifični oblici relacija koji zadovoljavaju određena pravila u vezi sa funkcionalnim zavisnostima
 - koji zato *garantuju* da u relaciji neće biti redundantnosti određenog tipa

Normalne forme

- U svom prvom radu o relacionom modelu (1970), Kod je definisao samo **1. normalnu formu (1NF)**
- Godinu dana kasnije (1971) je definisao **2. normalnu formu (2NF)** i **3. normalnu formu (3NF)**
- Zajedno sa Rejmondom Bojsom je definisao (1974) **Bojs-Kodovu normalnu formu (BCNF)**
- Kasnije je formulisano još normalnih formi

1. normalna forma

- Relacija je u prvoj normalnoj formi ako svaki atribut može da ima samo atomične vrednosti.
- Relacioni model pretpostavlja atomične vrednosti
- ER omogućava i složene vrednosti atributa

1. normalna forma (primer)

- Neka jedan predmet može da drži više nastavnika
- *Neatomični* model bi omogućavao da u relaciji *predmet* postoji atribut *nastavnici* sa imenima više nastavnika
- 1.NF to zabranjuje
- Jedno moguće rešenje je da se atribut *nastavnici* zameni većim brojem atributa nastavnik1, nastavnik2, nastavnik3
 - takvo rešenje nije dobro ni ako ima manje ni ako ima više od 3 nastavnika za neki predmet, pa čak ni kada ih je tačno 3
- Bolje rešenje je da se uvede nova relacija koja modelira odnos predmeta i nastavnika

2. normalna forma

- Relacija je u drugoj normalnoj formi
 - ako je u prvoj normalnoj formi i
 - ako nijedan neključan atribut (tj. koji ne pripada nijednom kandidatu ključu) nije funkcionalno zavisn od nekog pravog podskupa atributa nekog kandidat ključa.
- Suština je da u slučaju složenog ključa svi ostali atributi zavise od celog ključa a ne od nekog njegovog dela.
 - t.j. kandidati ključevi su minimalni

2. normalna forma (primer)

- Neka sledeća relacija opisuje predmete koje su upisali studenti:
UpisanPredmet{ **indeks**, **šifpred**, ime, prezime, nazivpred, kojiput }
- Relacija je u 1.NF
- Relacija nije u 2.NF
- ime i prezime zavise od atributa indeks
- nazivpred zavisi od atributa šifpred

Da bi se zadovoljila 2.NF potrebno je da se relacija podeli na više relacija

2. normalna forma (primer)

- Relaciju

UpisanPredmet{ **indeks**, **šifpred**, ime, prezime, nazivpred, kojiput }

Možemo da podelimo na 3 relacije:

- UpisanPredmet{ indeks, šifpred, kojiput }
- Student{ indeks, ime, prezime }
- Predmet{ šifpred, nazivpred }

- Prema teoremi o dekompoziciji, dekompozicija je potpuna

- Pokazuje se za jednu po jednu dekompoziciju, najpre:
 - UpisanPredmet{ **indeks**, **šifpred**, nazivpred, kojiput }
 - Student{ **indeks**, ime, prezime }
- a zatim i
 - UpisanPredmet{ **indeks**, **šifpred**, kojiput }
 - Predmet{ šifpred, nazivpred }

2. normalna forma (diskusija)

- Narušavanje 2.NF ukazuje na neispravno grupisanje atributa u relacije
 - relacija modelira više različitih FZ
 - to vodi redundantnosti
- Postizanje 2.NF je ključno za pozicioniranje atributa u *odgovarajuće* relacije
- Ipak, 2.NF dopušta da u relaciji postoje dva ključa kandidata, kao i tranzitivne zavisnosti
- Uvek je moguća potpuna dekompozicija do 2NF

3. normalna forma

- Ako je R relacija i X neki podskup njenih atributa i A neki atribut relacije R , onda je R u 3.NF akko za svaku FZ $X \rightarrow A$ na relaciji R važi jedno od:
 - $A \in X$, tj. A pripada skupu X (tzv. trivijalna zavisnost)
 - X je natključ (tj. ključ ili sadrži ključ)
 - A je deo nekog ključa relacije
- Originalna definicija (Kod, 1971)
 - Relacija je u 3.NF akko je
 - u 2.NF i
 - svaki neključan atribut je netranzitivno zavisan od svakog ključa.

3. normalna forma

- U osnovi nalaže da svaki neključni atribut može da predstavlja samo dodatni opis nekog ključa
- Tj. nema tranzitivnih zavisnosti među neključnim atributima (neključni atributi ne predstavljaju opise neključnih atributa)

3. normalna forma (primer)

- Neka sledeća relacija opisuje upisane studente:

Student{ **indeks**, ime, prezime, studprogram, nivo, trajanje }

- Relacija jeste u 2.NF ali nije u 3.NF
 - *nivo* i *trajanje* zavise od neključnog atributa studprogram
- Da bi se zadovoljila 3.NF potrebno je da se relacija podeli na 2 relacije:

Student{ **indeks**, ime, prezime, studprogram }

Program{ **studprogram**, nivo, trajanje }

3. normalna forma (diskusija)

- Narušavanje 3.NF ukazuje da je u relaciji modelirano više relativno nezavisnih FZ
- Uvek je moguća potpuna dekompozicija do 3.NF

Bojs-Kodova normalna forma

- Ako je R relacija i X neki podskup njenih atributa i A neki atribut relacije R , onda je R u Bojs-Kodovoj normalnoj formi akko za svaku FZ $X \rightarrow A$ na relaciji R važi jedno od:
 - $A \in X$, tj. A pripada skupu X , tzv. trivijalna zavisnost
 - X je natključ
- Tj.
 - svaki atribut relacije je ili deo ključa ili zavisi od **celog** ključa
 - ne postoje zavisnosti među neključnim atributima ni tranzitivne zavisnosti

Bojs-Kodova normalna forma, diskusija

- Relacije koje su u 3.NF najčešće zadovoljavaju i BKNF
- Praktično uvek kada postoji samo jedan ključ ili nema preklapanja među ključevima
- Teško je pronaći dobar primer koji
 - jeste u 3.NF
 - nije u BKNF
 - može da se prevede u BKNF

BKNF, primer 1.1

(https://en.wikipedia.org/wiki/Boyce-Codd_normal_form)

- Posmatramo relaciju

Rezervacija{ **teren**, **od**, do, tarifa }

- postoje dva terena, jedan je travnati i jedan betonski
- rezervacije su definisane terenom i periodom
- postoje 4 tarife – po dve za svaki od terena zavisno da li je termin predviđen za članove ili ne

- Relacija jeste u 2.NF i 3.NF (štaviše, nema atributa koji su van ključeva)

- postoji veći broj ključeva
 - { **teren**, **od**, do, tarifa }
 - { **teren**, od, **do**, tarifa }
 - { teren, **od**, do, **tarifa** }
 - { teren, od, **do**, **tarifa** }

- Relacija **nije** u BKNF

- problem je u zavisnosti tarifa -> teren
 - tarifa nije natključ
 - teren ne pripada skupu { tarifa }

BKNF, primer 1.2

- Jedno rešenje je dodavanje novog atributa *za_članove*:

Rezervacija{ **teren, od**, do, tarifa, za_članove }

- Sada postoje dodatne FZ:

- { tarifa } -> { teren, za_članove }
- { teren, za_članove } -> { tarifa }

- Koristimo drugu za potpunu dekompoziciju:

Rezervacija{ **teren, od**, do, za_članove }

Tarifa{ teren, za_članove, **tarifa** }

- može i obrnuto, da ključ bude tarifa, zato što je FZ u oba smera
- Ispostavlja se da je problem u **skrivenim** FZ koje postaju očigledne kada se dodaju novi atributi

BKNF i NFEK

- Neka imamo relaciju:

Prijavalspita(*student*, *ispitni rok*, predmet)

ključ je { student, ispitni rok }

- i zavisnosti:

- { student, ispitni rok } $\rightarrow$ { predmet }
 - tj. student u jednom isp.roku može da prijavi ispit iz najviše jednog predmeta
- { predmet } $\rightarrow$ { ispitni rok }
 - tj. svaki predmet se polaže samo u po jednom ispitnom roku

- Tada:

- relacija *Prijavalspita* nije u BKNF:
 - ispitni rok zavisi od predmeta, a predmet nije deo ključa
- ako napravimo dekompoziciju, narušićemo prvu zavisnost...
- ako promenimo ključ (*student*, *predmet*), problem opstaje

BKNF i NFEK

- Ali:
 - relacija Prijavalspita **jeste** u NFEK
 - prva zavisnost je od celog ključa
 - druga je prema delu ključa
 - obe su elementarne

BKNF i NFEK

- Postoji potpuna dekompozicija:
 - Prijavalspita(**student**, predmet)
 - PredmetURoku (**predmet**, ispitni rok)
- Ona je potpuna, ali ne čuva sve zavisnosti
 - nije očuvano ograničenje da u jednom ispitnom roku student može da prijavi ispit iz najviše jednog predmeta
- Štaviše, ne postoji dekompozicija koja čuva sve FZ

BKNF i NFEK

- Značaj NFEK je u tome što neke relacije ne mogu da pređu put od 3NF do BKNF a da se očuvaju sve zavisnosti...
- ...ali mogu da dođu bar do NFEK
- Da bi se u modelu sačuvala sve zavisnosti, potrebno je da se postojećoj relaciji doda nova relacija koja opisuje drugu zavisnost:
PredmetURoku (***predmet***, ispitni rok)
 - Uvodi se redundantnost...
 - ...koja se rešava dodavanjem stranog ključa (predmet, ispitni rok) iz relacije Prijavalspita prema relaciji PredmetURoku
 - (što ne podržavaju svi RSUBP, zato što se strani ključ ne vezuje za primarni)

Literatura za temu

- Teorey, Lightstone, Nadeau, Jagadish, Database Modeling and Design, 5.ed, Elsevier, 2011.
- Watt, Eng, Database Design, 2.ed, Open Edition, 2014.
- Gordana Pavlović-Lažetić, Uvod u relacione baze podataka, 2.izd. Matematički fakultet, 1999.
dostupno onlajn: <http://poincare.matf.bg.ac.rs/~gordana//urbp-2016.htm>
-
- Ramakrishnan, Gehrke, Database Management Systems, 2.ed, 2000.
- Darween, Date, Fagin, A Normal Form for Preventing Redundant Tuples in Relational Databases, ICDT '12 Proceedings, 2012.