

Univerzitet Crne Gore
Elektrotehnički fakultet

Prof. dr Vesna Popović-Bugarin

Logički dizajn baza podataka korišćenjem normalizacije

Logički dizajn baza podataka

- Faza unutrašnjeg (internog) dizajna baze podataka - uključuje dizajniranje „unutrašnjosti“ baze koju većina korisnika nikada neće vidjeti.
- Zadatak koji bi se trebao obavljati aplikacijom se dijeli u module (individualne jedinice programiranja aplikacije koje će biti napisane i testirane zajedno) i daje se detaljna specifikacija svakog modula.
- Sa stanovišta buduće baze podataka, **normalizacija** je najvažnij zadatak u fazi (logičkog) dizajna baze podataka.
 - Normalizacija je razvijena od strane Dr. E. F. Codd za dizajniranje tabela u relacionim bazama podataka, koje su najbolje za sisteme zasnovane na transakcijama (sistemi koji vrše dodavanje, ažuriranje i brisanje podataka u tabelama relacionih baza podataka).
 - Normalizacija nas uči kako da najbolje organizujemo podatke u tabelama.

Normalizacija baza podataka

- Normalizacija je proces pravljenja skupa relacija koji poseduje određeni skup osobina.
- Dr E. F. Codd je razvio proces normalizacije korišćenjem tri normalne forme, kasnije su se pojavile i druge normalne forme.

Nenormalizovana
relacija

Ukloniti redundantne attribute i
attribute sa više vrijednosti

Prva normalna
forma relacije

Ukloniti djelimično zavisne
attribute

Druga normalna
forma relacije

Ukloniti tranzitivno zavisne
attribute

Treća normalna
forma relacije

Dalja normalizacija?

Normalizovana
relacija

Logički dizajn - termini

- Kroz proces normalizacije ćemo koristiti logičke izraze za sve elemente baze.
- Podaci logički predstavljeni u dvodimenzionalnom formatu čine **relaciju – skup n-torki sa isti atributima.**
- Fizički dizajn baza podataka je zapravo taj koji vrši transformisanje normalizovanih relacija u relacione tabele i postoji određeno odstupanje u preslikavanju normalizovanih relacija u fizičke tabele.

Logički izraz	Fizički izraz
Relacija	Tabela
Jedinstveni identifikator	Primarni ključ
Atribut	Kolona
n-torka (tuple)	Vrsta

POTREBA ZA NORMALIZACIJOM

- U svom ranom radu o teoriji relacionih baza podataka, Dr Codd je otkrio da nenormalizovane relacije uvode određene probleme prilikom pokušaja ažuriranja podataka u njima.
- Za ove probleme Dr Codd je koristio izraz *anomalije*.
- Razlog normalizacije relacija je uklanjanje anomalija iz podataka.
- Ove anomalije su od suštinskog značaja za razumijevanje normalizacije, jer nam govore kada je prihvatljivo zaobići pravila prilikom fizičkog dizajna uz pomoć „denormalizacije“ relacija.
- Da bismo zaobišli pravila moramo razumjeti zašto ta pravila postoje.

NORMALIZACIJA KROZ PRIMJER

FIKT preduzeće RAČUN

Broj Kupca: 123674

Kupac: M. Marković

Ul. Dragana Radulovića 18

(00) 38267123-456

Uslovi: Net 15

Transport Via: USPS

Datum Porudžbine: 03/11/2017

Br. Proizvoda	Opis	Količina	Cijena	Suma
=====	=====	=====	=====	=====
ABC-2230	Dizne	3	50.00	150.00
DEF-25	Diferencijal za 220 Cdi	2	150.00	300.00
GHI-12	Hladnjak Vode	1	130.00	130.00
GHK-113	Hladnjak Klime	2	120.00	240.00
ACC-1	Desni Far	4	80.00	320.00

Ukupna cijena porudžbine:

€ 1,140.00

ANOMALIJE - INSERT

- *Anomalija insertovanja* predstavlja situaciju gdje se nova toraka (vrsta) ne može insertovati u relaciju zbog vještačke zavisnosti od druge relacije.
- Greška koja je uzrokovala anomaliju je ta da su atributi različitih entiteta pomiješani u istu relaciju.
- Vidimo da su ID, ime i adresa kupca uključeni u račun (pogled).
- Ako bismo željeli da napravimo relaciju iz ovog pogleda kakav jeste, a nakon toga i tabelu iz relacije, otkrili bismo da se novi kupac ne može insertovati u bazu ukoliko nešto nije kupio. Ovo je stoga što svi podaci vezani za kupca uključeni u sam račun.

ANOMALIJE - DELETE

- *Anomalija brisanja* je suprotna anomaliji insertovanja.
- Odnosi se na situaciju gdje brisanje podataka vezanih za određeni entitet uzrokuje neplanirani gubitak podataka koji karakteriše drugi entitet.
- U slučaju našeg računa, ako odlučimo da obrišemo posljednji (jedini preostali) račun koji pripada određenom kupcu, izgubićemo sve podatke vezane za tog kupca.
- Opet, ovo je zbog toga što su dva entiteta (kupci i računi) nepravilno kombinovani u jednu relaciju, jer je račun implementiran u vidu tabele bez primjene procesa normalizacije na relaciju.

ANOMALIJE - UPDATE

- *Anomalija ažuriranja* se odnosi na situaciju gde ažuriranje jednog podatka zahtijeva ažuriranje više n-torki (vrsta) podataka.
- U primjeru našeg računa, ako bismo htjeli da promijenimo adresu kupca, morali bismo je promijeniti na svakom računu tog kupca. Ovo je zbog toga što bi adresa kupca bez primjene normalizacije bila redundantno smještena na svakom računu kupca.
- Da bi stvari bile gore, redundantni podaci znače ažuriranje više kopija tih podataka, a ovo dalje znači da bi neažuriranje nekih kopija dovelo do nekonzistentnosti podataka.
- Zbog toga se koristi pravilo po kome se vrijednost svakog atributa smješta na jednom mestu i ta se kopija koristi svuda.

PRIMJENA PROCESA NORMALIZACIJE

- Proces normalizacije se primjenjuje na svaki korisnički pogled prikupljen tokom ranijih faza dizajna.
- Započinjemo sa svakim korisničkim pogledom kao relacijom, što znači da ga predstavljamo kao da je dvodimenzionalna tabela.
- Tokom procesa normalizacije dolaziće do prepisivanja postojećih relacija i kreiranja novih.

Broj Kupca	Ime Kupca	Adresa Kupca	Grad Kupca	Država Kupca	Zip Kod Kupca	Telefon Kupca	Uslovi	Transport Via	Datum Porudžbine	Br. Poizvoda	Opis	Količina	Cijena	Suma
123674	M. Marković	Ul. Dragana Radulovića 18	Podgorica	MNE	81000	38267-123-456	Net 15	USPS	3/3/2011	ABC-2230	Dizne	3	50.00	150.00
										DEF-25	Diferencijal za 220 Cdl	2	150.00	300.00
										GHI-12	Hladnjak Vode	1	130.00	130.00
										GHK-113	Hladnjak Klime	2	120.00	240.00
										ACC -1	Desni Far	4	80.00	320.00
										Ukupna cijena porudžbine			€ 1,140.00	

PRIMJENA PROCESA NORMALIZACIJE

- Višestruke vrijednosti su smještene u ćelijama kolona koje sadrže podatke (opis) za stavke računa.
- Ovakve attribute nazivamo *atributima sa više vrijednosti*, jer kao što im ime kaže, mogu imati više vrijednosti za neke n-torke (vrste) u relaciji.
- Ako bismo napravili bazu podataka na ovaj način, mogućnost da koristimo SQL ili neki drugi jezik za pravljenje upita tim kolonama bila bi vrlo ograničena:
 - Pronalaženje svih porudžbina koje sadrže određeni proizvod zahtijevalo bi da pretražujemo podatke kolone pomoću operatora LIKE. Ažuriranje bi bilo jednako teško.
 - Brisanje jednog proizvoda sa računa bi zahtijevalo primjenu naredbe UPDATE umesto DELETE, jer ne želimo da obrišemo čitav račun.

PRIMJENA PROCESA NORMALIZACIJE

Broj Kupca	Ime Kupca	Adresa Kupca	Grad Kupca	Država Kupca	Zip Kod Kupca	Telefon Kupca	Uslovi	Transport Via	Datum Porudžbine	Br. Poizvoda	Opis	Količina	Cijena	Suma
123674	M. Marković	Ul. Dragana Radulovića 18	Podgorica	MNE	81000	38267-123-456	Net 15	USPS	3/3/2011	ABC-2230	Dizne	3	50	150
123674	M. Marković	Ul. Dragana Radulovića 18	Podgorica	MNE	81000	38267-123-456	Net 15	USPS	3/3/2011	DEF-25	Diferencijal za 220 Cdl	2	150	300
123674	M. Marković	Ul. Dragana Radulovića 18	Podgorica	MNE	81000	38267-123-456	Net 15	USPS	3/3/2011	GHI-12	Hladnjak Vode Hladnjak	1	130	130
123674	M. Marković	Ul. Dragana Radulovića 18	Podgorica	MNE	81000	38267-123-456	Net 15	USPS	3/3/2011	GHK-113	Hladnjak Klime	2	120	240
123674	M. Marković	Ul. Dragana Radulovića 18	Podgorica	MNE	81000	38267-123-456	Net 15	USPS	3/3/2011	ACC -1	Desni Far	4	80	320
											Ukupna cijena porudžbine			€ 1,140.00

PRIMJENA PROCESA NORMALIZACIJE

- Ovde su kolone sa više vrednosti smeštene u odvojene vrste, dok su podaci iz drugih kolona ponovljeni potreban broj puta.
- Očiti problem u ovom pristupu je ponavljanje podataka. Tako se na primer, ime i adresa kupca ponavljaju za svaku stavku računa, što ne samo da je nepotrebno trošenje resursa, već nas takođe izlaže nekonzistenciji kada god se podaci ne održavaju na isti način (na primer, ažuriramo ime grada za jednu stavku, a preskočimo za ostale).

PRIMJENA PROCESA NORMALIZACIJE

- Prepisivanje korisničkih pogleda u tabele sa svim potrebnim podacima je mukotrpan i spor proces.
- Koristićemo sljedeći pristup po kome sve atribute navodimo u vidu liste i zamišljamo da su dvodimenzionalne tabele.

RAČUN: Broj Kupca, Ime Kupca, Adresa Kupca, Grad Kupca, Država Kupca, Zip Kod Kupca, Telefon Kupca, Uslovi, Transport Via, Datum Porudžbine, Br. Proizvoda, Opis, Količina, Cijena, Suma, Ukupna Cijena Porudžbine

- Radi jasnoće dodato je ime relacije, pri čemu je ono navedeno velikim slovima i odvojeno od atributa dvotačkom.

ODABIR PRIMARNOG KLJUČA

- Prvi korak u normalizaciji je odabir primarnog ključa između svih jedinstvenih identifikatora koje pronađemo u relaciji.
- Podsjetimo se da jedinstveni identifikator predstavlja jedan ili kolekciju više atributa koji jedinstveno identifikuju svaku pojavu (n-torku) u relaciji.
- Vrlo često je to samo jedan atribut.
- Ukoliko jedinstveni identifikator ne može biti u vidu jednog atributa, možemo nadovezati nekoliko atributa da formiramo identifikator.
- Važno je razumjeti da kada je jedinstveni identifikator sastavljen od više atributa, sami atributi se ne kombinuju, tj. oni i dalje postoje kao nezavisni atributi i postaće pojedinačne kolone u tabeli, ili tabelama, kreiranoj iz naših normalizovanih relacija.

ODABIR PRIMARNOG KLJUČA

- Ponekad, ali rijetko, dešava se da ne postoji smislen skup atributa u relaciji koji može poslužiti kao jedinstveni identifikator. Ipak, kada se ovo desi, moramo izmisliti jedinstveni identifikator, često sa vrijednostima dodijeljenim sekvencijalno ili slučajno, tokom dodavanja instance entiteta bazi podataka.
- Ova tehnika predstavlja izvor jedinstvenih identifikatora kao što su ID zaposlenih, broj tablica vozila, broj motora vozila, itd.
- Jedinstvene identifikatore koji imaju stvarno - smisleno značenje nazivamo *prirodnim* identifikatorima, dok ove druge (uključujući one koje moramo izmisliti) nazivamo *surogatnim* ili *vještačkim* identifikatorima.

ODABIR PRIMARNOG KLJUČA

Broj Kupca	Ime Kupca	Adresa Kupca	Grad Kupca	Država Kupca	Zip Kod Kupca	Telefon Kupca	Uslovi	Transport Via	Datum Porudžbine	Br. Poizvoda	Opis	Količina	Cijena	Suma
123674	M. Marković	Ul. Dragana Radulovića 18	Podgorica	MNE	81000	38267-123-456	Net 15	USPS	3/3/2011	ABC-2230	Dizne	3	50.00	150.00
										DEF-25	Diferencijal za 220 Cdl	2	150.00	300.00
										GHI-12	Hladnjak Vode	1	130.00	130.00
										GHK-113	Hladnjak Klime	2	120.00	240.00
										ACC -1	Desni Far	4	80.00	320.00
											Ukupna cijena porudžbine		€ 1,140.00	

- *Broj Kupca?*
- *Broj Kupca kombinovan sa datumom?*

PRIMARNI KLJUČ - RAČUN

- U našem primjeru, *Broj Kupca* na računu jedinstveno identifikuje podatke o kupcu unutar računa, ali pošto kupac može imati više računa, nije adekvatno izabrati broj kupca kao identifikator za čitav račun.
- Možemo pokušati da iskombinujemo broj kupca sa datumom porudžbine, ali ako bi kupac imao dva računa sa istim datumom, gubimo jedinstvenost identifikatora.
- U primjeru našeg računa imamo slučaj da ne postoji prirodni identifikator za relaciju.
- Otuda je mnogo bolje izmisliti identifikator, kao što je na primer *Broj Računa*.

PRIMARNI KLJUČ - Odabir

- Moramo imati na umu da izabrani identifikator mora *uvijek* biti jedinstven, tj. ako postoji bar jedan slučaj gdje on nije jedinstven, ne možemo ga koristiti.
- Primjer loše odabranog primarnog ključa je kombinacija: imena i prezimena; očevog i majčinog imena i datuma rođenja (brazilska vlade za registrovanje glasača 1994.)...
- Ponekad relacija ima više od jednog mogućeg jedinstvenog identifikatora. Kada se ovo desi, svaku mogućnost nazivamo *kandidatom*. Kada smo identifikovali sve moguće kandidate za relaciju, moramo izabrati jednog kandidata koji će biti primarni ključ relacije.
- Odabir primarnog ključa je *suštinski* za proces normalizacije, jer sva pravila normalizacije koriste primarni ključ.

PRIMARNI KLJUČ - Odabir

- Kriterijumi za odabir primarnog ključa među kandidatima su (prvo ide najvažniji):
 - *Ako postoji samo jedan kandidat, izaberi ga.*
 - *Odaberi onog kandidata za kojeg je najmanje vjerovatno da će promijeniti svoju vrednost.*
 - *Odaberi najjednostavnijeg kandidata.* Onaj kandidat koji je sastavljen od najmanjeg broja atributa se smatra najjednostavnijim.
 - *Odaberi najkraćeg kandidata.* Ovo je čisto radi efikasnosti. Ako se primarni ključ može pojaviti u mnogim tabelama kao spoljašnji ključ, često je dobro da se sačuva malo prostora na ovaj način.

PRIMARNI KLJUČ – Račun

- U primjeru našeg računa, izabrali smo vještački primarni ključ nazvan Broj Računa.
- Na ovaj način dobijamo jednostavan primarni ključ za sve račune naše kompanije koji je garantovano jedinstven, jer DBMS-u možemo naložiti da automatski dodjeljuje sekvencijalne brojeve novim računima prilikom njihovog kreiranja.
- Što se tiče označavanja primarnog ključa u našem zapisu relacije u vidu liste, usvojen je karakter taraba (#) ispred imena atributa (jednog ili više) primarnog ključa.

RAČUN: #Broj Računa, Broj Kupca, Ime Kupca,
Adresa Kupca, Grad Kupca, Država Kupca,
Zip Kod Kupca, Telefon Kupca, Uslovi,
Transport Via, Datum Porudžbine, Br.
Proizvoda, Opis, Količina, Cijena, Suma,
Ukupna Cijena Porudžbine

PRVA NORMALNA FORMA: *ELIMINISANJE PONAVLJANJA PODATAKA*

- Za relaciju se kaže da se nalazi u *prvoj normalnoj formi* (PNF) kada ne sadrži attribute sa više vrijednosti.
- Ponekad ćemo naići na grupu atributa koji se ponavljaju, kao što su pojedinačne stavke našeg računa. Svaki atribut u grupi ima više vrijednosti, ali su neki atributi toliko srodni da se njihove vrijednosti zajedno ponavljaju.
- Na ovaj način dobijamo tzv. *ponavljajuću grupu*, ali u suštini, to je ništa drugo do specijalan slučaj problema atributa sa više vrijednosti.

PNF - Račun

- Po konvenciji ćemo ponavljajuće grupe i atribute sa više vrijednosti grupisati u zagradama.

RAČUN: #Broj Računa, Broj Kupca, Ime Kupca,
Adresa Kupca, Grad Kupca, Država Kupca,
Zip Kod Kupca, Telefon Kupca, Uslovi,
Transport Via, Datum Porudžbine, (Br.
Proizvoda, Opis, Količina, Cijena, Suma),
Ukupna Cijena Porudžbine

- Podaci nekog kupca će se ponavljati za svaki njegov račun, ali tim problemom ćemo se pozabaviti kada dođemo do treće normalne forme.
- Iako postoji mnogo kupaca ove firme, uvek je samo jedan kupac pridružen jednom računu, tako da podaci kupca na računu ne predstavljaju ponavljajuću grupu.
- Pošto postoji samo jedan kupac po računu, problem ponavljanja podataka na svim računima kupca se neće razmatrati prilikom transformacije relacije u prvi normalni oblik.

PNF

- Da bismo transformisali nenormalizovanu relaciju u prvu normalnu formu moramo premjestiti sve attribute sa više vrijednosti i ponavljajuće grupe u nove relacije.
- Pošto je ponavljajuća grupa skup atributa koji se *zajedno* ponavljaju, svi atributi u ponavljajućoj grupi treba da se premjeste u jednu novu relaciju.
- Sa druge strane, pojedinačni atributi sa više vrijednosti treba da se premjeste u njima dodijeljene nove relacije, a ne da se kombinuju sa drugim atributima sa više vrijednosti u jednoj relaciji. Kao što ćemo vidjeti kasnije, ovakvim pristupom izbjegavamo probleme četvrte normalne forme.

PNF - algoritam

1. Kreirajte novu relaciju, po mogućnosti sa smislenim imenom.
2. Kopirajte primarni ključ iz originalne u novu relaciju. Ovaj kopirani primarni ključ sad postaje spoljašnji ključ na originalnu relaciju.
3. Premjestite ponavljajuću grupu ili atribut sa više vrijednosti u novu relaciju.
4. Učinite primarni ključ (kopiran iz originalne relacije) jedinstvenim dodavanjem atributa iz ponavljajuće grupe. Ako premještate atribut sa više vrijednosti, dodajte baš taj atribut primarnom ključu.
5. Opciono, možete izabrati da zamijenite primarni ključ sa jednim surogatnim ključem, zadržavajući attribute koji sačinjavaju prirodni primarni ključ formiran u koracima 2. i 4.

PNF - Račun

RAČUN: #Broj Računa, Broj Kupca, Ime Kupca, Adresa Kupca, Grad Kupca, Država Kupca, Zip Kod Kupca, Telefon Kupca, Uslovi, Transport Via, Datum Porudžbine, Ukupna Cijena Porudžbine

STAVKA NA RAČUNU: #Broj Računa, #Br. Proizvoda, Opis, Količina, Cijena, Suma

- Atribut *Broj Računa* je kopiran iz *RAČUN* u *STAVKA NA RAČUNU* i *Br. Proizvoda* mu je dodat da bi se formirao primarni ključ relacije *STAVKA NA RAČUNU*.
- Čitava ponavljajuća grupa (*Br. Proizvoda, Opis, Količina, Cijena, Suma*) je uklonjena iz relacije *RAČUN*.
- Ne postoje ponavljajuće grupe ili atributi sa više vrijednosti u relacijama, tako da se te relacije nalaze u prvoj normalnoj formi.

PNF - Račun

- Postoji interesantna posljedica formiranja prirodnog primarnog ključa za relaciju *STAVKA NA RAČUNU*: Ne možemo staviti isti proizvod na dati račun više od jedanput u smislu da se u listi proizvoda svaki proizvod može naći tačno jednom.
- Ovo može biti poželjno, ali može biti i ograničavajući faktor našoj kompaniji. Da bismo znali šta je od to dvoje slučaj, moramo razumjeti poslovna pravila firme.
- Alternativa je da napravimo surogatni primarni ključ za relaciju *STAVKA NA RAČUNU*.
- U nastavku ćemo koristiti već formiran „prirodni“ ključ, dobijen kombinacijom dva atributa.

Funkcionalna zavisnost

- Atribut *B* je funkcionalno zavisan od atributa *A* ako u bilo kom trenutku ne postoji više od jedne vrijednosti atributa *B* koja odgovara datoj vrijednosti atributa *A*.
- Ako kažemo da je atribut *B* funkcionalno zavisan od atributa *A*, na taj način takođe kažemo da atribut *A* *određuje* (*determiniše*) atribut *B* ili da je *A* *determinant* (jedinostveni identifikator) atributa *B*.

RAČUN: #Broj Računa, Broj Kupca, Ime Kupca,
Adresa Kupca, Grad Kupca, Država Kupca,
Zip Kod Kupca, Telefon Kupca, Uslovi,
Transport Via, Datum Porudžbine, Ukupna
Cijena Porudžbine

- Činjenica da *Broj Računa* jedinstveno određuje *Broj Kupca* u ovoj relaciji znači da je *Broj Kupca* funkcionalno zavisan od *Broj Računa*

DRUGA NORMALNA FORMA: *ELIMINISANJE DJELIMIČNIH ZAVISNOSTI*

STAVKA NA RAČUNU: #Broj Računa, #Br. Proizvoda,
Opis, Količina, Cijena, Suma

- Atribut Opis je funkcionalno zavisan od atributa *Br. Proizvoda*
- Za relaciju se kaže da se nalazi u *drugoj normalnoj formi* (DNF) ako zadovoljava oba sljedeća kriterijuma:
 - Relacija se nalazi u prvoj normalnoj formi.
 - Svi ne-ključni atributi su funkcionalno zavisni od **čitavog** primarnog ključa.
- Opis je funkcionalno zavisan od **dijela** primarnog ključa, tj. samo od *Br. Proizvoda*, a ne i od kombinacije *Broj Računa* i *Br. Proizvoda*.

DNF - Račun

- Bilo koja relacija u prvoj normalnoj formi koja ima jedan atribut za primarni ključ, automatski se nalazi u drugoj normalnoj formi.

STAVKA NA RAČUNU: #Broj Računa, #Br. Proizvoda,
Opis, Količina, Cijena, Suma

- *STAVKA NA RAČUNU* ne zadovoljava drugu normalnu formu po gore pomenutom automatizmu.
- *Br. Proizvoda* sam određuje vrijednost atributa *Opis* i *Cijena*, a ne kombinacija atributa *Broj Računa* i *Br. Proizvoda*
- Kada u relaciji pronađemo narušavanje druge normalne forme, rješenje je da premjestimo atribut(e) koji je djelimično zavisian u novu relaciju, gdje će zavisiti od čitavog ključa umjesto od njegovog dijela.
- Cijena?

DNF - Račun

RAČUN: #Broj Računa, Broj Kupca, Ime Kupca,
Adresa Kupca, Grad Kupca, Država Kupca,
Zip Kod Kupca, Telefon Kupca, Uslovi,
Transport Via, Datum Porudžbine, Ukupna
Cijena Porudžbine

STAVKA NA RAČUNU: #Broj Računa, #Br. Proizvoda,
Količina, Prodajna Cijena, Suma

PROIZVOD: #Br. Proizvoda, Opis, Trenutna Cijena

- Održavanje *Opis-a* sada nema anomalija. Možemo uvesti novi proizvod nezavisan od računa vezanog za taj proizvod. Ako želimo da promijenimo Opis, to radimo jednostavno promjenom jedne vrijednosti u jednoj vrsti podataka. Ako želimo da obrišemo i posljednji račun za određeni proizvod iz baze, nećemo izgubiti njegov opis, tj. on će se i dalje naći u vrsti relacije PROIZVOD.

TREĆA NORMALNA FORMA: *ELIMINISANJE TRANZITIVNIH ZAVISNOSTI*

- Za atribut koji zavisi od drugog atributa koji nije primarni ključ relacije se kaže da je **tranzitivno zavisan**.

RAČUN: #Broj Računa, Broj Kupca, Ime Kupca,
Adresa Kupca, Grad Kupca, Država Kupca,
Zip Kod Kupca, Telefon Kupca, Uslovi,
Transport Via, Datum Porudžbine, Ukupna
Cijena Porudžbine

- Za relaciju se kaže da se nalazi u trećoj normalnoj formi (TNF) ako zadovoljava oba od sljedeća dva kriterijuma:
 - Relacija se nalazi u drugoj normalnoj formi.
 - Ne postoji tranzitivna zavisnost (tj. svi ne-ključni atributi zavise samo od primarnog ključa).

TNF - Račun

- Da bismo transformisali relaciju koja se nalazi u drugoj normalnoj formi u treću normalnu formu potrebno je premjestiti sve tranzitivno zavisne atribute u relacije gdje će zavisiti samo od primarnog ključa.
- Pri ovome je potrebno ostaviti atribut od kojeg zavise u originalnoj relaciji da služi kao spoljašnji ključ, koji će nam trebati za rekonstrukciju originalnog korisničkog pogleda kroz povezivanje tabela.

RAČUN: #Broj Računa, Broj Kupca, Uslovi,
Transport Via, Datum Porudžbine, Ukupna
Cijena Porudžbine

KUPAC: #Broj Kupca, Ime Kupca, Adresa Kupca, Grad
Kupca, Država Kupca, Zip Kod Kupca, Telefon
Kupca

TNF - Račun

RAČUN: #Broj Računa, Broj Kupca, Uslovi,
Transport Via, Datum Porudžbine, Ukupna
Cijena Porudžbine

STAVKA NA RAČUNU: #Broj Računa, #Br. Proizvoda,
Količina, Prodajna Cijena, Suma

PROIZVOD: #Br. Proizvoda, Opis, Trenutna Cijena

KUPAC: #Broj Kupca, Ime Kupca, Adresa Kupca, Grad
Kupca, Država Kupca, Zip Kod Kupca, Telefon
Kupca

- Ukupna Cijena Porudžbine? Suma?
- Određeni kombinacijom drugih ne-ključnih atributa i samim tim i tranzitivno zavisni od njih.
- TNF nalaže da uklonimo attribute koji se jednostavno izračunavaju.

TNF – Račun – konačan

RAČUN: #Broj Računa, Broj Kupca, Uslovi,
Transport Via, Datum Porudžbine

STAVKA NA RAČUNU: #Broj Računa, #Br. Proizvoda,
Količina, Prodajna Cijena

PROIZVOD: #Br. Proizvoda, Opis, Trenutna Cijena

KUPAC: #Broj Kupca, Ime Kupca, Adresa Kupca, Grad
Kupca, Država Kupca, Zip Kod Kupca, Telefon
Kupca

- Pravila prve, druge i treće normalizacije se lako mogu sumirati kao: **U relaciji koja se nalazi u trećoj normalnoj formi svaki atribut treba da zavisi od primarnog ključa, cijelog primarnog ključa i ničega osim primarnog ključa.**
- Zip kod kupca?
- Anomalije?

NAKON TNF

- Treća normalna forma pokriva više od 90% slučajeva koji se srijeću kod poslovnih informacionih sistema i smatra se „zlatnim standardom“ u poslovnim sistemima.
- Kad se ovlada trećom normalnom formom vrijedi pogledati i dodatne normalne forme.

BOYCE-CODD-OVA NORMALNA FORMA

ČETVRTA NORMALNA FORMA

PETA NORMALNA FORMA

DOMEN – KLJUČ NORMALNA FORMA (DKNF)

BOYCE-CODD-OVA NORMALNA FORMA

- Boyce-Codd-ova normalna forma (BCNF) predstavlja jaču verziju treće normalne forme.
- Odnosi se na anomalije koje se javljaju kada je ne-ključni atribut determinant atributa koji je dio primarnog ključa, tj. kada je atribut koji je dio primarnog ključa funkcionalno zavisian od ne-ključnog atributa (postoji samo jedna vrijednost atributa koji je dio ključa koja odgovara datoj vrijednosti ne-ključnog atributa).
- Kao primjer, pretpostavimo da naša firma dodjeljuje svakom kupcu specijalistu koji pruža pomoć vezanu za **jednu** određenu liniju proizvoda.

BCNF

Ime Kupca	Linija Proizvoda	Specijalista za Podršku
M. Marković	Dizne	I. Ivanović
M. Marković	Diferencijal	P. Petrović
M. Marković	Hladnjak	B. Brnović
M. Janković	Far	D. Dragović
D. Durutović	Far	D. Dragović
D. Durutović	Diferencijal	P. Petrović
P. Petrović	Hladnjak	B. Brnović
P. Petrović	Far	D. Dragović

- Moramo kombinovati *Ime Kupca* i *Linija Proizvoda* da bismo formirali primarni ključ.
- *Specijalista za Podršku* određuje atribut *Linija Proizvoda*.
- Surogatni primarni ključ bi narušio treću normalnu formu, jer bi ne-ključni atribut (*Specijalista za Podršku*) određivao drugi ne-ključni atribut (*Linija Proizvoda*).
- Ovde smo maskirali grešku normalizacije praveći *Linija Proizvoda* dijelom primarnog ključa.

BCNF

- Ispunjenje Boyce-Codd-ove normalne forme zahtijeva sljedeće:
 - Relacija mora biti u trećoj normalnoj formi.
 - Ne postoje determinanti izuzev primarnog ključa ili kandidata. Drugim rečima, ne-ključni atribut ne može jedinstveno identifikovati bilo koji drugi atribut, uključujući i onaj koji predstavlja dio primarnog ključa.
- Rješenje se sastoji u podjeli neželjenog determinanta u dvije tabele, kao što bismo uradili kada se naruši treća normalna forma. BCNF verzija ove relacije data je ispod.

DODJELA SPEC. ZA PODRŠKU: #Broj Kupca, #ID Spec. za Podršku

SPECIJALNOST SPEC. ZA PODRŠKU: #ID Spec. za
Podršku, Specijalizacija

BCNF

Ime Kupca	Specijalista za Podršku
M. Marković	I. Ivanović
M. Marković	P. Petrović
M. Marković	B. Brnović
M. Janković	D. Dragović
D. Durutović	D. Dragović
D. Durutović	P. Petrović
P. Petrović	B. Brnović
P. Petrović	D. Dragović

Specijalista za Podršku	Linija Proizvoda
I. Ivanović	Dizne
P. Petrović	Diferencijal
B. Brnović	Hladnjak
D. Dragović	Far
D. Dragović	Far
P. Petrović	Diferencijal
B. Brnović	Hladnjak
D. Dragović	Far

IV NORMALNA FORMA

- Dodatna anomalija se pojavljuje kada su dva ili više atributa sa više vrijednosti uključeni u istu relaciju.
- Pretpostavimo na primjer, da želimo da vodimo evidenciju istovremeno o administrativnim i jezičkim vještinama naših zaposlenih. Možemo doći do relacije:

ID Zaposlenog	Administrativne Vještine	Jezik
5001	Kuca 30rpm	Engleski
5001	Excel	Francuski
5002	Daktilograf	Ruski
5002	Excel	Francuski

- Za ovu relaciju možemo formirati primarni ključ biranjem kombinacije bilo *ID Zaposlenog i Administrativne Vještine* ili *ID Zaposlenog i Jezik*.

VJEŠTINE ZAPOSLENOG: #*ID Zaposlenog*,
#*Administrativne Vještine*, *Jezik*

VJEŠTINE ZAPOSLENOG: #*ID Zaposlenog*, #*Jezik*,
Administrativne Vještine

IV NORMALNA FORMA

- Pokazuje se da obje alternative, pored treće normalne forme, zadovoljavaju i Boyce-Codd-ovu normalnu formu.
- Problem, naravno, leži u tome da postoji implicitna veza između administrativnih i jezičkih veština.
- Striktnim pridržavanjem procedura prve normalne forme može se izbjeći ovaj problem.
- Ipak, ako se sretne narušavanje četvrte normalne forme, rješenje je jednostavno i sastoji se u premještanju svih atributa sa više vrijednosti u posebne relacije:

```
AD VJEŠTINE ZAPOSLENOG: #ID Zaposlenog,  
                        #Administrativne Vještine
```

```
VJEŠTINE ZAPOSLENOG: #ID Zaposlenog, #Jezik
```

PETA NORMALNA FORMA

- Neki autori i istraživači su sugerisali potrebu za petom normalnom formom koja se bavi specijalnim tipom ograničenja poznatog kao *zavisnost spajanja*, za čije razumijevanje je potrebno znanje relacionog računa.
- Drugi su opisali petu normalnu formu na potpuno isti način kao što je ovde opisana četvrta normalna forma.
- Ukratko, ne postoji jasna standardna definicija pete normalne forme u industriji, i dok zavisnosti spajanja mogu biti od teorijskog interesa, ne postoji jasna evidencija da one imaju praktičnu vrijednost u poslovnim aplikacijama.

DENORMALIZACIJA

- Normalizacija vodi ka novim relacijama koje se dalje transliraju u nove tabele i spajanja.
- Kada korisnici baze imaju problema sa performansama koji se ne mogu riješiti drugim sredstvima, kao što su podešavanje baze ili poboljšanje hardvera na kojem radi RDBMS, možda je jedino rešenje **denormalizacija**.
- Denormalizacija je danas daleko manje potrebna nego u ranim danima relacionih baza podataka.
- Denormalizacija nije isto što i izbjegavanje da se uradi normalizacija. Jednom kad se postigne normalizovan dizajn baze podataka, podešavanja se mogu napraviti imajući na umu potencijalne posljedice (anomalije).

DENORMALIZACIJA

- Mogući koraci denormalizacije uključuju sledeće:
 - Rekombinacija podijeljenih relacija sa zadovoljenjem pravila normalizacije.
 - Smještanje redundantnih podataka u tabele.
 - Smještanje sumarnih podataka u tabele.
- Normalizacija se koristi za uklanjanje anomalija iz baza koje se koriste za online transakciono-procesirajuće sisteme.
- Baze u kojima se smješta istorijat podataka i koje se koriste isključivo u analitičke svrhe, nisu predmet anomalija insertovanja, ažuriranja i brisanja.

LITERATURA

- **Andy Oppel: *Databases DeMYSTiFieD*, 2nd Edition.**